

Rapport d'activités hebdomadaire

Semaines 1 et 2 :
14/04 au 18/05
28/04 au 02/05

Nom : Ghislain Loaec

I- Activités de lecture (Articles de recherche, livrable de projets de recherche....)

- Towards topic maps for a promise theory based configuration management - Kefialew Eskedar
- Promise theory-a model of autonomous objects for pervasive computing and swarms – Burgess Mark 2006
- Laws of human-computer behaviour and collective organization – Burgess Mark 1999
- A process algebra based framework for promise theory – Bergstra 2007
- In search of an understandable consensus algorithm - Ongaro Diego 2013
- Revisiting the Paxos algorithm - De Prisco Roberto 1997
- Paxos made simple – Lamport 2001

II- Activités de rédaction (Etat de l'art, Articles de recherches....)

Prises de notes à la lecture des articles ci-dessus : [cf. Notes_02_06.pdf](#)

III- Activités de développement (Framework, Simulation, Implémentation...)

Création du dépôt pour le projet en vue d'une preuve par implémentation :

<https://github.com/gloaec/trifle>

Création d'un store de configuration basé sur une ontologie à l'aide de la librairie python « rdflib ». Ce magasin d'informations détiendra des données de contexte (données sondées et promesses) et de configuration (clé → valeur) dans l'infrastructure. Ces données seront formalisées à l'aide d'une ontologie sous format RDF (plus riche) ou N3 (plus lisible). Le choix de TopicMaps n'est pas retenu à cause de son impopularité et du manque de banques d'information.

Consultation de la documentation de la librairie « gitpython », choix technologique potentiel pour la gestion du versionning des ontologies. L'outil de versionning « git » permettrait un approche simplifiée dans la gestion des conflicts lors d'une fusion des ontologies et imposerait une certaine rigueur dans l'édition collaborative par le système multi-agent.

Esquisse d'interface web pour :

- La visualisation de l'activité des différents agents
- La consultation des données de contexte et de configuration

- Le renseignement de politiques de gestion et de directives d'administration

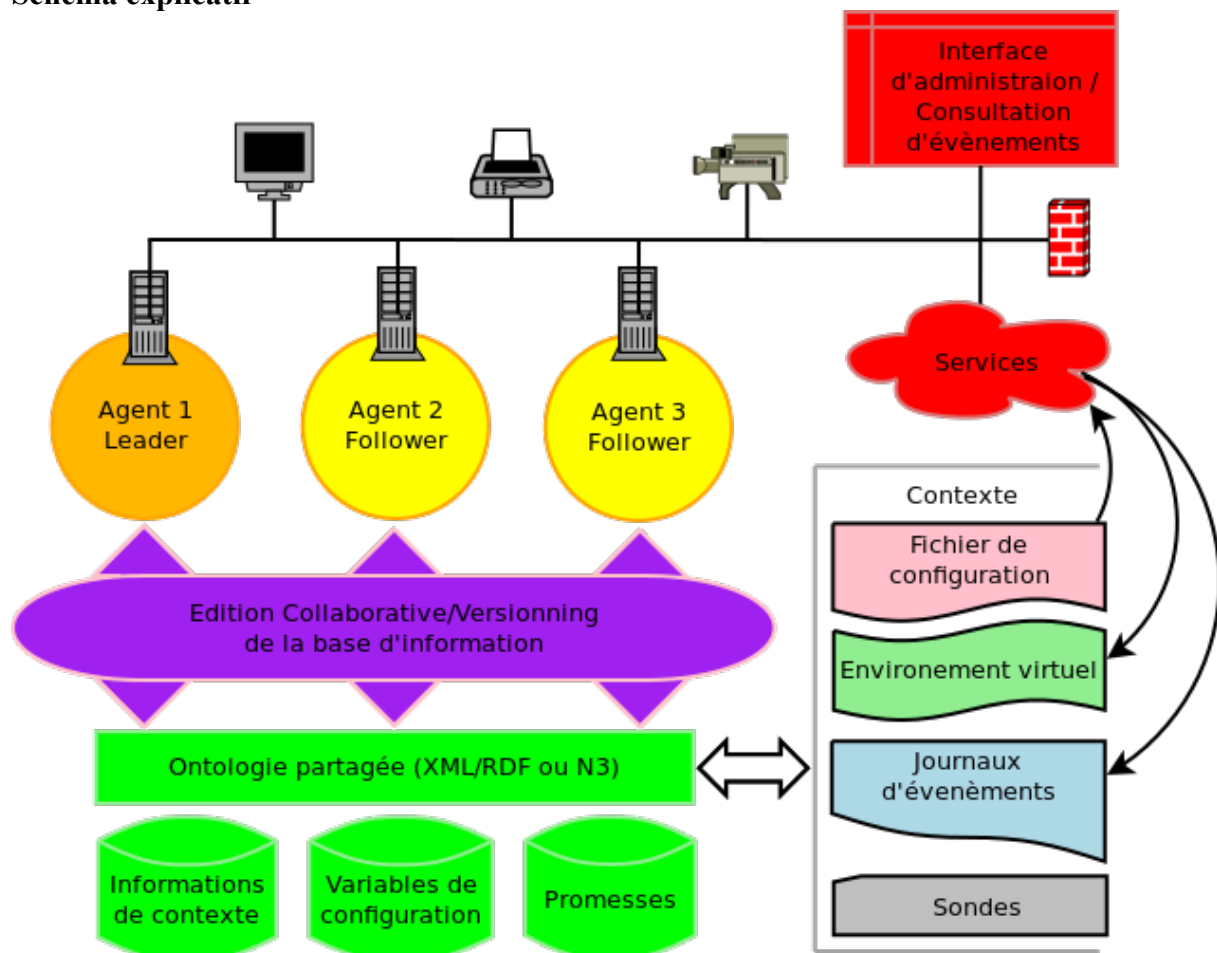
Les politiques d'administration se traduisent par des directives sémantiques permettant un niveau d'abstraction plus ou moins élevé du paramétrage global dans l'architecture. L'exécution de ces directives s'appuierait sur un ensemble de « recettes » réunissant les procédures et les promesses nécessaires à la mise en place des services désirés. Afin de garantir une tolérance maximum à l'échec :

- l'ontologie devra être la plus exhaustive possible ;
- l'écriture des recettes nécessitera un haut degré de formalisation ;
- un consensus (raft ou paxos) doit être mis en place.

Le consensus de raft n'est pas implémenté, il serait intéressant de jeter un coup d'oeil à <https://github.com/coreos/etcd>

La structure actuelle prévoit une structure bien distincte des exceptions. Une interface en ligne de commande est implémentée et l'interface web permet en l'état d'effectuer des requêtes SPARQL brutes sur l'ontologie.

Schéma explicatif



Problématiques émergentes :

Comment synchroniser les informations entre les différents agents de l'infrastructure ?

Comment assurer la communication inter-agents à moindre effort (zéro configuration) ?

Quel protocole utiliser ?

Comment garantir la pérennité de cette liaison ?

Avec l'utilisation de « git », comment gérer le serveur de dépôt ?

Comment répliquer/synchroniser l'interface web de configuration.

IV- Autres Activités (Présentations, Cours école doctorale....)