Université de Montréal

Modèle de confiance et ontologie probabiliste pilotés par réseaux bayésiens pour la gestion des accords de services dans l'environnement de services infonuagiques

Par
Obed JULES

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des arts et des sciences
en vue de l'obtention du grade de Maîtrise ès sciences (M.Sc.)
en informatique

Août 2014

Résumé

L'infonuage est un nouveau paradigme de services informatiques disponibles à la demande qui a connu une croissance fulgurante au cours de ces dix dernières années. Le fournisseur du modèle de déploiement public des services infonuagiques décrit le service à fournir, le prix, les pénalités en cas de violation des spécifications à travers un document. Ce document s'appelle le contrat de niveau de service (SLA). La signature de ce contrat par le client et le fournisseur scelle la garantie de la qualité de service à recevoir. Ceci impose au fournisseur de gérer efficacement ses ressources afin de respecter ses engagements.

Malheureusement, la violation des spécifications du SLA se révèle courante, généralement en raison de l'incertitude sur le comportement du client qui peut produire un nombre variable de requêtes vu que les ressources lui semblent illimitées. Ce comportement peut, dans un premier temps, avoir un impact direct sur la disponibilité du service. Dans un second temps, des violations à répétition risquent d'influer sur le niveau de confiance du fournisseur et sur sa réputation à respecter ses engagements.

Pour faire face à ces problèmes, nous avons proposé un cadre d'applications piloté par réseau bayésien qui permet, premièrement, de classifier les fournisseurs dans un répertoire en fonction de leur niveau de confiance. Celui-ci peut être géré par une entité tierce. Un client va choisir un fournisseur dans ce répertoire avant de commencer à négocier le SLA. Deuxièmement, nous avons développé une ontologie probabiliste basée sur un réseau bayésien à entités multiples pouvant tenir compte de l'incertitude et anticiper les violations par inférence. Cette ontologie permet de faire des prédictions afin de prévenir des violations en se basant sur les données historiques comme base de connaissances.

Les résultats obtenus montrent l'efficacité de l'ontologie probabiliste pour la prédiction de violation dans l'ensemble des paramètres SLA appliqués dans un environnement infonuagique.

Mots-clés **:** Infonuage, SLA, réseau bayésien, ontologie probabiliste, répertoire, infonuagique, modèle de confiance.

Abstract

Cloud Computing is a new paradigm of IT on-demand services which has experienced tremendous growth over the past decade. The provider of Cloud computing services describes the service to be provided, its cost, and penalties for service violations within a document. This document is called Service Level Agreement (SLA). The signature of this contract by the customer and the provider guarantees the quality of service received by the customer. It also entails the provider to manage its resources efficiently to meet its commitments.

Unfortunately, the SLA violation is common; it is usually caused by uncertainty about customer behavior that can make variable number of requests assuming that resources are boundless. This behavior may have an impact on the availability of the service, thus its related SLA. Repeated SLA violations will definitively have an impact on the trust level that the customer has about the provider that might no longer enjoys a good reputation in meeting its commitments.

To cope with these problems, we have proposed a Framework driven by a Bayesian network that allows, first, to classify the suppliers in a Cloud directory according to their trust level. This directory can be managed by a third party entity, in which a client will choose a provider before starting SLA negotiation. Secondly, we have developed a probabilistic ontology, based on a Multi-Entity Bayesian network, which takes into account uncertainty, in the customer behavior, and makes predictions by inference; these predictions help preventing SLA violations based on historical data..

The results show the effectiveness of the probabilistic ontology for the prediction of SLA violations in a Cloud Computing environment.

Keywords: Cloud computing, SLA, Bayesian network, probabilistic ontology, Cloud-directory, trust model.

Table des matières

Liste des tableaux

Liste des figures

Liste des acronymes

CRM	Customer Relationship Management
EC2	Elastic Cloud Compute
BNT	Bayesian Net Toolbox
FoSII	Foundations of Self-governing ICT Infrastructures
IaaS	Infrastructure as a Service
JSP	Java Server Pages
LoM2HiS	Low-level Metric to High-level SLA
LPD	Local Probability Distribution
MATLAB	Matrix Laboratory
MEBN	Multi-Entity Bayesian Network
MFrag	MEBN Fragment
MTBF	Mean Time Between Failure
MTheory	MEBN Theory
MTTR	Mean Time to Repair
MV	Machine Virtuelle
NIST	National Institute of Standards and Technology
OWL	Web Ontology Language
PaaS	Plateform as a Service
QdS	Qualité de service
QoS	Quality of Services
RDF	Ressource Description Framework
RDF	Resource Description Framework
RDFS	RDF Schema
RIF	Rules interchange Format
ROC	Receiver Operating Characteristic
S3	Simple Storage Service
SaaS	Software as a Service
SGBD	Système de gestion de base de données
SIG	Système d'information géospatiale
SLA	Service Level Agreement
SLO	Service Level Objective

SPARQL	SPARQL Protocol and RDF Query Language)
SWRL	Semantic Web Rule Language
UCT	Unité centrale de traitement
URI	Uniform Resource Identifier
VM	Virtual Machine
W3C	World Wide Web Consortium
XML	Extensible Markup Language

À toi Théo…

Remerciements

Mes sincères remerciements à mon directeur de recherche, professeur Hafid Abdelhakim, ainsi qu'à mon co-directeur de recherche, professeur Mohamed Adel Serhani. Je vous remercie pour vos soutiens et conseils que vous m'avez offerts tout au long de mes recherches.

Merci au programme canadien de bourse de la francophonie (BCBF), pour avoir financé ces deux années d'études. Un grand merci spécialement à la gestionnaire de ce programme Madame Jeanne Gallagher pour ses conseils et soutiens.

Un mot spécial à mes parents qui ont toujours cru en moi. Merci pour ses appels téléphoniques et vos mots d'encouragements. Merci pour tous ces sacrifices que vous avez consentis pour mon éducation.

Enfin, comment ne pas remercier ma vaillante épouse? Merci à toi pour ton courage, tu as été là chaque fois que j'en avais besoin, tu m'as épaulé, tu m'as soutenu dès le début jusqu'à la fin. Mille mercis!

Chapitre 1 : Introduction

Au cours de ces dix dernières années, l'Internet s'est imposé comme un espace attrayant pour le développement des services informatiques accessibles à la demande. Plusieurs facteurs innovants (p.ex., l'omniprésence du haut débit, les standards et les normes garantissant l'interopérabilité des logiciels et des capacités de stockage de plus en plus performants) offrent un cadre propice à cet environnement. L'un des derniers nés de ces services c'est l'infonuage (*Cloud Computing, angl.*)(Chieu, et al. 2009). La technologie de l'infonuage permet d'offrir aux entreprises et aux particuliers des ressources informatiques en couches dématérialisées disponibles à la demande moyennant un modèle de paiement à travers l'Internet (d'où la notion de *nuage*). La dématérialisation des ressources joue un rôle majeur dans la croissance de l'infonuage, car elle permet aux entreprises, en particulier, de développer leurs solutions informatiques sans avoir à recourir à des investissements préalables à la mise en place de l'infrastructure physique du système d'une part, et sans se soucier de la gestion des supports physiques et logiques des couches voisines (réseaux, sécurité, etc.) d'autre part. Ainsi, l'infonuage se démarque de l'architecture client/serveur et l'administration du parc informatique traditionnel dans les entreprises, mais se veut garant d'une accessibilité ubiquitaire du service, peu importe le support utilisé.

Le *National Institute of Standards and Technology (*NIST), agence du département du commerce des États-Unis définit l'infonuage comme : « *Un modèle permettant un accès pratique, ubiquitaire et à la demande à un ensemble commun de ressources informatiques (tels que réseaux, serveurs, stockage, applications et services) configurables disponibles rapidement et sans efforts de gestion ni interactions avec le fournisseur de services. [NIST — 800 — 145]* ». (Hogan, et al. 2011). La technologie infonuagique est basée sur la gestion des ressources composées essentiellement de centre de données sur lesquels sont configurées de nombreuses machines virtuelles (VM) pour fournir un service. La gestion de ces centres de données doit garantir :

- **La redondance des informations et de la configuration des ressources**. Cela implique des centres de données *(Datacenter, angl.)* interopérables répartis sur des sites différents afin de mieux gérer les temps de maintenance.
- **Une maintenance concurrentielle**. Une redondance efficace des informations et des ressources garantit une gestion optimale des planifications de maintenance dans un délai qui n'influe pas sur la disponibilité des services.
- **Tolérances aux failles et robustesse**. Afin d'assurer un taux élevé de la disponibilité du service, aucune panne n'est tolérée en dehors du temps imparti à la maintenance.

Le NIST (Hogan, et al. 2011) définit aussi les cinq caractéristiques essentielles à l'infonuage :

- **Service à la demande :** un client peut unilatéralement s'approvisionner des ressources informatiques du nuage de manière automatique sans interaction humaine.
- **Large accès réseau :** le service doit être accessible via les standards Internet, de n'importe où, peu importe le support matériel utilisé par l'utilisateur.
- **Mutualisation des ressources :** les ressources physiques et virtuelles doivent être assignées et configurées de manière à répondre automatiquement à la demande du client avec une abstraction de haut niveau sur la localisation physique des ressources.
- **Élasticité et souplesse :** du point de vue de l'utilisateur, les ressources doivent sembler illimitées et doivent être adaptables au besoin de façon automatique et rapide. C'est-à-dire, les ressources doivent pouvoir s'élargir ou se rétrécir en fonction du comportement du client. Ce dernier peut s'approvisionner en payant le coût unitaire des ressources, au besoin, à tout moment et en quantité illimitée. C'est la mise en échelle du service.
- **Service mesurable :** le service doit être calculé selon un modèle de paiement transparent entre les parties avec des règles bien définies sur le coût associé à l'utilisation des ressources ainsi que les pénalités qui en découlent. Ce calcul de coup droit être automatique, pour des ressources illimitées « *Pay As You Go !* », en fonction des règles établies et le coût par unité de chaque type de service **(p.ex., stockage, bande passante, processeur et mémoire)**

Les différentes ressources (matériels et logiciels) de l'infonuage sont divisées en couches qui peuvent être offertes en tant que services indépendants selon le modèle de service en question (voir la figure 2). Le NIST (Hogan, et al. 2011) distingue donc trois principaux modèles de services :

- Logiciel en tant que service (SaaS, pour Software as a Service), permet de rendre accessible des logiciels pour des utilisateurs sans que ces derniers n'aient pas à se soucier de la gestion des systèmes d'exploitation ou de la gestion des paramètres

réseau. Le SaaS offre un service applicatif comme les calendriers, courriels et bureautique.

- Plateforme en tant que service (PaaS, pour Platform as a Service), permet de rendre disponible des environnements de développement, des librairies et des systèmes de bases de données.
- Infrastructure en tant que service (IaaS, pour Infrastructure as a Service) est la capacité à fournir au client des ressources fondamentales telles qu'un espace de stockage, réseaux, virtualisation avec la liberté de choisir son système d'exploitation, système de gestion de base de données (SGBD) et intégration.

La mise en place d'une architecture de l'infonuage suit un modèle de déploiement qui peut être privé, public, hybride ou communautaire (Bauer and Adams 2012). Dans le cadre de cette recherche, nous avons mis l'accent sur le modèle de déploiement public. Pour comprendre le rôle de chaque intervenant dans les services infonuagiques, le NIST (Hogan, et al. 2011) définit quatre principaux acteurs :

1. Le fournisseur de service infonuagique qui peut-être :
    - o un fournisseur d'infrastructure en tant que service ('IaaS) : qui offre des services d'infrastructures de base (réseaux, VM, stockage, système d'exploitation, mémoire, hébergement).
    - o un fournisseur de programme en tant que service (PaaS) : qui développe des solutions applicatives, gère des intergiciels (*middleware*, *angl*.) et offre des outils de développement avec la possibilité de rendre disponible un ensemble d'interfaces de programmation applicative (API) pour les développeurs. Par contre, celui-ci n'a pas à se soucier de la configuration du réseau, mémoire, VM, etc… .
    - o un fournisseur de logiciel en tant que service (SaaS) : qui installe et gère des applications (p.ex., bureautique, statistiques, systèmes d'information géospatiale (SIG) et courriels) qu'il offre en tant que service. Ce fournisseur n'a pas à gérer les logiciels médiateurs encore moins le système d'exploitation.

2. Le consommateur infonuagique comprend deux grandes catégories **:**
    - o le client : est celui (une personne ou une entreprise) qui achète les services d'un fournisseur et pouvant développer des applications et vendre des services à son tour. Par exemple, un fournisseur SaaS peut être un client pour un fournisseur PaaS de même que ce dernier peut être un client pour le fournisseur IaaS qui lui, peut être un client pour un fournisseur de service d'Internet sur IP.
    - o les utilisateurs finaux : sont ceux qui utilisent les applications pour accomplir des tâches et des activités spécifiques sans avoir besoin de connaissance particulière sur la configuration de l'infrastructure du système.

3. Le courtier (*broker*) : est une partie tierce, il est chargé de la promotion d'un ou plusieurs fournisseurs. Il gère l'utilisation, la performance, la livraison du service au client, et négocie les relations entre les fournisseurs infonuagiques et les clients. Un client peut donc commander un service d'infonuage directement chez le fournisseur ou en passant par le courtier.
4. Le transporteur : est une entité intermédiaire qui transporte le service infonuagique et assure la connectivité entre le client et le fournisseur. C'est le rôle joué par un fournisseur d'accès d'Internet par exemple.
5. L'auditeur : est une entité tierce qui peut procéder à une évaluation et/ou la mise en œuvre indépendante de système d'exploitation, de sécurité et de test de performance pour les services infonuagiques.

L'entente entre un client et un fournisseur se matérialise via un contrat, appelé l'accord de niveau de service (SLA : *Service Level Agreement,* angl.) (Sun, et al. 2010). Le SLA est décrit plus en détail dans la section 1.1 de ce chapitre. Dans ce contrat, le fournisseur définit les règles, la description, et les paramètres du service à offrir ainsi que le coût associé par unité consommée ainsi que les pénalités encourues en cas de non-respect de certaines clauses.

Ce document présente le résultat des recherches sur l'application du SLA dans l'environnement infonuagique public. Nous avons analysé les principales avancées ayant été proposées dans la littérature sur le sujet afin d'y apporter notre contribution.

1.1. Motivation

Aujourd'hui, le marché infonuagique connait une croissance considérable et continue encore à progresser. D'après le magazine économique Gartner (Newsroom 2013) l'infonuage public, à l'entremise de l'infrastructure en tant que service (IaaS), vaut 131 milliards de dollars américains sur le plan d'investissement à travers le monde en 2013. L'investissement dans l'infonuage en 2012, toujours selon Gartner (Newsroom 2013), était de 111 milliards de dollars. L'infonuage public a connu donc une augmentation

d'investissement de 18.5 % en 2013 par rapport à 2012. Gartner (Newsroom 2013) annonce une projection pour 2016 où l'infonuage à travers le monde vaudra 677 milliards de dollars d'investissement. D'ici là, il y a des standards à mettre en place pour réduire les pertes et augmenter la disponibilité du service dans le respect du SLA.

Le SLA joue un rôle important dans l'environnement infonuagique, car il est un garant des services à obtenir du point de vue du client. Mais aussi, il impose au fournisseur de bien gérer ses ressources afin de respecter les engagements qui y sont spécifiés.

Dans la spécification d'un SLA, on peut clairement identifier au moins trois parties (Wieder, et al. 2011) : la partie 1 identifie les antagonistes impliqués dans la signature du contrat *(par exemple Fournisseur-client)* et décrit le rôle de chacun d'entre eux et la durée du contrat*;* la partie 2 décrit les paramètres du contrat comme les unités de mesure des ressources, les indicateurs composites, coûts, sécurité et lieux; et la partie 3 qui décrit les objectifs de niveau de service (SLO : *Service Level Objectives*). Ces derniers sont des caractéristiques mesurables du SLA (Zhang, et al. 2010), comme la disponibilité du service, le temps de réponse et la bande passante. Un SLO permet d'évaluer la performance du fournisseur par la QdS (qualité de service – *QoS : Quality of Service,* angl.) offerte sous forme de seuils (une disponibilité supérieure à 95 % par exemple). Dans (Sturm, et al. 2000) Rick Sturm et coll. décrivent les caractéristiques que doivent rencontrer les SLOs; elles doivent être : atteignables, mesurables, compréhensibles, significatives, contrôlables, abordables, et mutuellement acceptables.

Les fournisseurs de service infonuagique doivent spécifier dans le SLA tous les SLOs. Un des paramètres importants du SLO est la disponibilité du service; par exemple, la plupart des fournisseurs du marché offrent une disponibilité mensuelle de 99.95 % (Gagnaire, et al. 2012) , soit 21 minutes de temps d'arrêt mensuel du service. Cumulée au cours de l'année, cette valeur du temps d'arrêt non soumis aux pénalités vaut environ 4 heures. Toutefois, certains facteurs peuvent empêcher au fournisseur de fournir une bonne qualité de service par rapport à la disponibilité. Certains de ces facteurs peuvent être externes à l'infrastructure infonuagique, n'impliquant pas directement le fournisseur, mais plutôt le client. Par exemple, une connexion internet de mauvaise qualité du côté du client ou un vol d'identité peuvent rendre l'accès au service impossible par ce dernier. Par contre,

d'autres sont internes à l'infrastructure infonuage, impliquant directement le fournisseur du service. Par exemple, une panne du service Amazon EC2 survint en avril 2011 qui avait duré plusieurs jours (Hagen, et al. 2012). Dans ce cas, les violations à répétition peuvent donc influer négativement sur la réputation du fournisseur.

La violation du SLA par le fournisseur peut être due à une gestion inefficace des ressources, mais aussi par l'incertitude qui règne dans l'environnement infonuagique. Ces incertitudes viennent du comportement du matériel qui peut tomber en panne malgré les dispositions mises en place, mais surtout du comportement des clients. Ce comportement insaisissable peut se caractériser par le nombre de requêtes, la mobilité du client dans des régions géographiques différentes, les téléchargements à taille plus ou moins variable, qui pourraient impliquer des besoins en ressources supplémentaires. L'application du SLA dans l'environnement infonuagique est un problème complexe qui doit tenir compte de la nature dynamique de l'infonuage. En plus, l'incertitude qui existe autour des infrastructures du système de l'infonuage et l'incertitude sur le comportement du client empêchent au fournisseur de respecter les clauses du contrat à la lettre.

Un fournisseur qui est incapable de gérer adéquatement ses ressources et incapable de cerner les incertitudes autour du service, risque de ne pas respecter ses engagements. Tandis que les violations à répétitions conduisent systématiquement à une mauvaise réputation. Il est donc devenu impossible à un nouveau client de choisir un fournisseur qui a un niveau de confiance élevé, jouissant d'une bonne réputation et offrant une perspective d'avenir rassurant en ce qui a trait au respect de ses engagements.

La meilleure approche pour résoudre ce problème consisterait à anticiper les cas de violations en faisant des prédictions par des méthodes de raisonnement capable de tenir compte des incertitudes.

1.2. Problématique

L'application de l'accord de niveau de service dans un environnement dynamique comme l'infonuage est un défi complexe. Elle exige, au prime abord une gestion efficace

des ressources disponibles afin d'éviter les violations des engagements consignés dans le contrat. La gestion efficace des ressources est problématique à cause les incertitudes au niveau du fonctionnement des équipements et les incertitudes sur le comportement du client.

Un fournisseur n'ayant pas la capacité de bien gérer ses ressources afin de respecter ses engagements, risque d'avoir une mauvaise réputation et un faible niveau de confiance auprès de ses clients. Un niveau de confiance faible soulève des doutes sur la capacité de ce fournisseur à garantir le respect de ses engagements à l'avenir.

Constatant que les violations sont dues principalement par une mauvaise gestion des ressources et ne tenant pas compte des incertitudes. Constatant que les violations diminuent le niveau de confiance du fournisseur. Dans le cadre de notre travail de recherche sur l'application du SLA dans l'environnement des services infonuagiques, nous avons opté à répondre à ces deux principales questions.

1. *Comment un nouveau client peut-il choisir un fournisseur de confiance jouissant d'une bonne réputation et qui soit en mesure de respecter ses engagements en offrant une qualité de service satisfaisante dans un futur proche?*
2. *Comment un fournisseur peut-il prévenir des violations du contrat de niveau de service en tenant compte des incertitudes?*

1.3. Contributions

Pour répondre aux deux questions posées dans la problématique, nous avons analysé les principales contributions déjà apportées sur le sujet afin de relever leurs faiblesses. Nous avons proposé une solution qui est axée sur deux grands points : (1) afin de faciliter, à un nouveau client, le choix d'un fournisseur de confiance, nous avons proposé un modèle de confiance basé sur une inférence bayésienne capable d'analyser les données historiques pour donner un score de confiance à un fournisseur. Ensuite, les fournisseurs sont classés dans un répertoire, géré par une entité tierce, par ordre de leur score de confiance; un nouveau client peut accéder à ce répertoire pour choisir un fournisseur avant de négocier la signature du SLA; (2) Pour prévenir les violations du SLA

dans un environnement infonuagique, nous avons implémenté une ontologie probabiliste, basée sur un réseau bayésien avec entités multiples (MEBN : *Multi-Entity Bayesian Network*) de premier ordre comme moteur d'inférence, capable de prendre en compte les incertitudes. Ce moteur d'inférence prend aussi en compte les données historiques du système comme base de connaissances afin d'inférer la probabilité des évènements pouvant conduire à une violation. L'analyse des requêtes sur l'ontologie développée montre l'efficacité de notre approche par rapport aux ontologies déterministes.

Les résultats de nos contributions ont fait l'objet d'un article qui été accepté dans le cadre de la 3e conférence internationale infonuagique de IEEE (*2014 IEEE 3rd International Conference on Cloud Networking (CloudNet) CLOUDNET'14*). Article ayant pour titre : « *Bayesian Network, and Probabilistic Ontology Driven Trust Model for SLA Management of Cloud Services* » dont les co-auteurs sont : Obed Jules, Abdelhakim Hafid et Mohamed Adel Serhani.

1.4. Organisation du mémoire

Ce rapport de mémoire est organisé de la manière suivante :

Le chapitre 1 présente une mise en contexte du sujet avec la définition et compositions de l'infonuage ainsi qu'une présentation de nos contributions. Le chapitre 2 présente une analyse critique de l'état de l'art des principales avancées et propositions faites pour la gestion automatique des ressources infonuagiques, la gestion de confiance et détection de violation du SLA. Le chapitre 3 décrit l'architecture du cadre d'applications que nous avons proposé. Le chapitre 4 est consacré à la présentation des approches d'implémentation permettant la collecte de données et l'expérimentation du cadre d'applications. Le chapitre 5 présente et analyse les résultats obtenus de l'expérimentation. En conclusion, le chapitre 6 rappelle des éléments de réponse à la problématique posée, les problèmes rencontrés au cours de cette recherche et identifie les perspectives de recherche d'avenir.

Chapitre 2 : État de l'art

Dans cette section, nous avons présentons les principaux concepts nécessaires à la compréhension de notre recherche. Nous avons analysé trois des concepts clés servant de piliers à notre recherche : la gestion des ressources infonuagiques dans la section (2.1), les modèles de confiance dans la section (2.2) et les cadres d'applications de gestion de SLA basé sur des ontologies dans la section (2.3).

2.1 Gestion des ressources

Le fournisseur de service Infonuagique doit gérer efficacement ses ressources afin d'offrir une QdS à ses clients qui soit conforme aux spécifications du SLA. Cette gestion doit garantir la prévention de violation du SLA. Or, un des défis majeurs de cette gestion est : comment correspondre les paramètres de bas niveau, comme *temps d'arrêt* ou *temps de marche*, de l'infrastructure physique, en des paramètres mesurables de haut niveau, comme la *disponibilité*, exprimés dans SLA?

2.1.1 Correspondance des paramètres de bas niveau en mesure de haut niveau

LoM2HiS (*Low-level Metric to High-level SLA*), est un cadre d'applications proposé par Emeakaroha et coll. (Emeakaroha, et al. 2010; Emeakaroha, et al. 2012; Maurer, et al. 2012) qui, comme son nom l'indique, permet de faire correspondre les paramètres de bas niveau des ressources disponibles en mesure de haut niveau comme les paramètres du SLA. LoM2HiS est basé sur le respect des objectifs fixés par la *Foundations of Self-governing ICT Infrastructures (FoSII).* FoSII est un projet de recherche, basé à l'Université de Vienne de technologie, qui propose un modèle et des concepts pour l'application et la gestion automatique du SLA (Michael Maurer 2012). L'infrastructure FoSII est utilisée afin de gérer le cycle de vie et l'autoadaptation du service Infonuagique. Sans intervention humaine, le système doit pouvoir s'adapter au changement et au comportement interne qu'externe (p.ex., changement de matériel, temps d'arrêt et panne au niveau des applications) de l'environnement du service infonuagique.

2.1.1.1 L'infrastructure FoSII

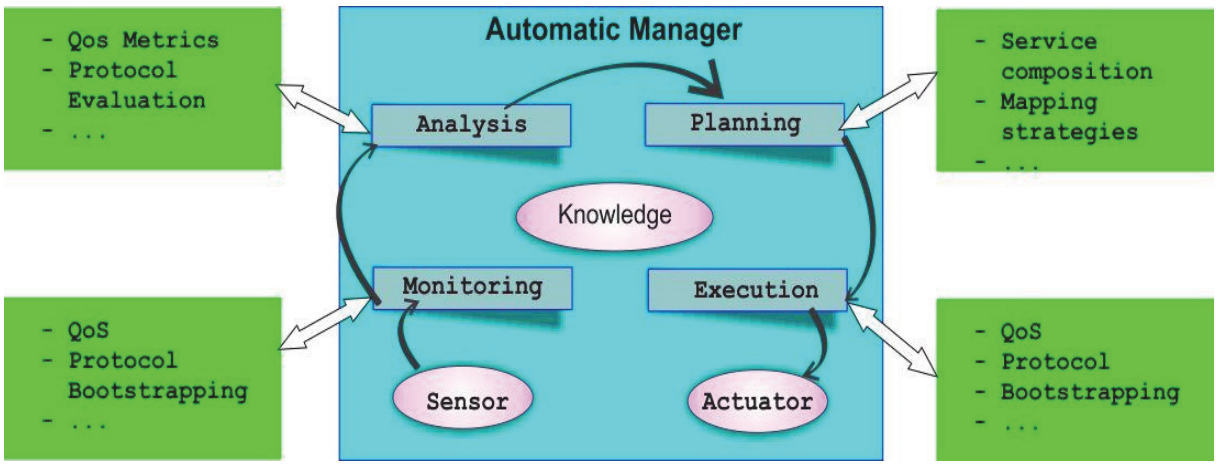


Figure 1 : Composition de l'infrastructure FoSII (Michael Maurer 2012).

Chaque service de l'infrastructure FoSII (Figure 1) implémente trois interfaces (i) interface de négociation nécessaire à l'établissement du contrat SLA; (ii) interface de gestion de tâches nécessaire au lancement des tâches; (iii) interface d'autogestion nécessaire à la prévention des violations. Pour exécuter ces fonctions, le modèle utilise des senseurs afin de détecter les états et le comportement du système. De point de vue logique, l'infrastructure FoSII contient deux parties : (i) le composant Enactor (voir figure 2) qui est responsable de l'autogestion du service et la connaissance de l'environnement; et (ii) LoM2HiS qui est responsable de la correspondance et du contrôle des ressources pour le composant Enactor**.**

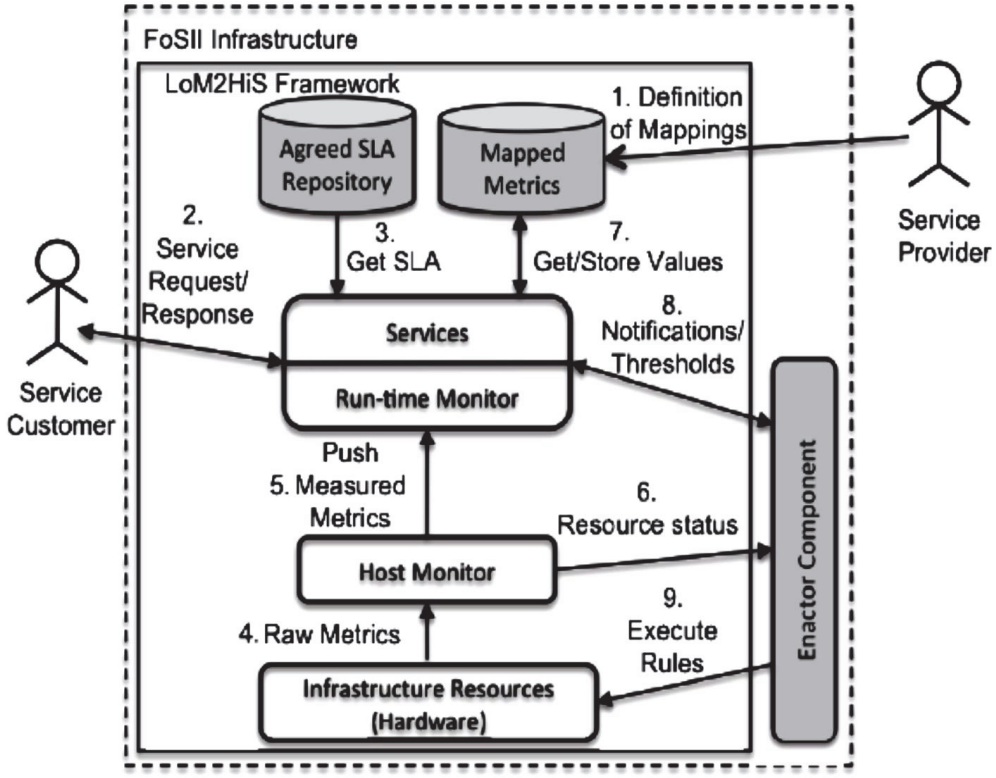


Figure 2 : Architecture du cadre d'applications LoM2HiS (Emeakaroha, et al. 2010)

2.1.1.2 Le Cadre d'applications LoM2HiS

La figure 4 présente l'architecture du cadre d'applications LoM2HiS basé sur l'infrastructure FoSII pour faire le contrôle des ressources et la correspondance entre les paramètres de bas niveau en mesures de haut niveau. Les différents composants et le fonctionnement de cette architecture ont utilisé un modèle de communication principalement centré autour de l'unité de gestion et de contrôle (*Run-time Monitor*). Ce centre de contrôle représente la couche applicative où le service est déployé; il est aussi responsable de contrôler le service sous la base des règles du SLA. Après avoir conclu la négociation du SLA, le fournisseur crée les règles de correspondance exploitable par LoM2HiS (étape 1 dans la figure 4). Le cadre d'applications LoM2HiS extrait les métriques brutes concernant les ressources (p.ex., matériel, réseaux et MV (étape 4) pour les calculer et utiliser au travers du centre de contrôle (*Run-Time Monitor*) tout en vérifiant l'état du système communiqué par le senseur du modèle FoSII (étape 6). En recevant les mesures calculées, le *Run-time Monitor* assure la correspondance en couple composé de

paramètres de bas niveau et les règles associées afin de trouver leur équivalent dans le SLO du SLA.

Le composant Enactor du modèle FoSII a pour rôle de communiquer les nouveaux seuils de ressources disponibles au centre de contrôle afin de prévenir des violations (étape 8) pendant le temps d'exécution du service. Selon l'état du système et le seuil de ressources disponibles, le composant *Enactor* peut définir de nouvelles règles afin d'en faire des ajustements (p.ex., allouer plus de mémoire à la VM numéro 10) (étape 9).

Dans ce paragraphe nous allons mettre l'accent sur le processus de correspondance des paramètres de bas niveau en mesures de haut niveau du SLA. L'équipe dirigeant le projet LoM2HiS (Emeakaroha, et al. 2010; Emeakaroha, et al. 2012; Maurer, et al. 2012) endosse au *Run-time Monitor* la responsabilité de la correspondance des mesures en deux types de règles. (i) les règles de correspondances simples qui font une correspondance un-à-un des mesures de bas niveau aux mesures de haut niveau (p.ex., espace disque disponible — stockage); et (ii) les règles de correspondance complexes qui utilisent des formules prédéfinies pour calculer les mesures de haut niveau du SLA en utilisant les mesures de bas niveau; le tableau I présente quelques-unes de ces règles de correspondances.

Tableau I : Tableau de correspondances des règles de correspondance complexe

Mesures des ressources (bas niveau)	**Paramètres du SLA (haut niveau)**	**Règles de correspondance**
Temps d'arrêt, *Temps de marche*	Disponibilité (D)	$D=100\% * \frac{MTBF}{MTBF+MTTR.}$
débitEntrant, *débitSortant,* *TailleDuPaquet,* *bandePassanteEntrant,* *bandePassanteSortant*	Temps de réponse (T_{total})	$T_{total} = T_{Rép\_R} + T_{Rép\_S}\ (ms)$
Temps moyen entre pannes (MTBF)	MTBF	$= \frac{\sum(TempsDeMarche - TempsD'Arrêt)}{Nombre\ de\ pannes}$

Les règles de correspondance complexe pour la disponibilité du service sont ainsi calculées : le paramètre de bas niveau « temps d'arrêt » est représenté par la mesure de haut niveau « MTTR » (*Mean Time to Repair*). MTTR est le temps écoulé entre le moment d'arrêt du système et le moment où le système recommence à fonctionner normalement. Le temps de marche est représenté par MTBF (*Mean Time Between Failure*); cette mesure est le temps moyen entre deux pannes; plus spécifiquement, elle représente le temps écoulé entre le moment où le système commence à fonctionner et le moment où une prochaine panne survienne.

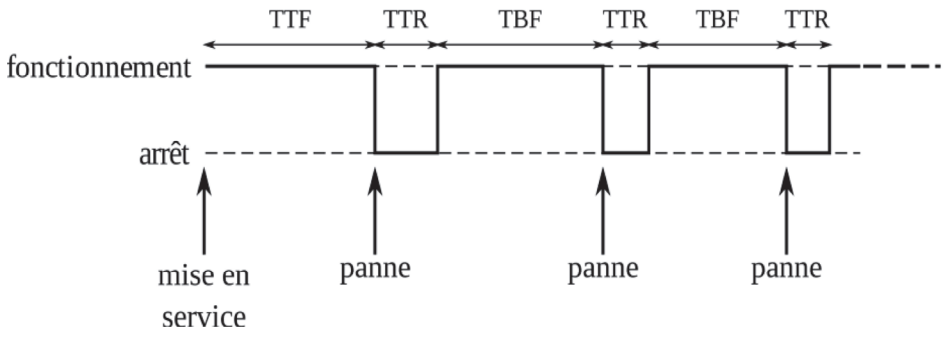


Figure 3 : Schéma explicatif du MTTR et du MTBF

Le temps de réponse est le temps émis par le système pour l'exécution d'une requête; il est composé, du temps subi par la requête en partant de l'émetteur (soit un client) vers la destination (soit le serveur) est noté $T_{Rép\_R}$ (Temps de réponse rentrant) et le temps subi par la réponse à la requête, noté $T_{Rép\_S.}$ Les deux sont exprimés en millisecondes et calculés de la manière suivante :

$$\boldsymbol{T_{Rép_S}} = \frac{TailleDuPaquet}{BandePassanteEntrant - débitEntrant}$$

$$\boldsymbol{T_{Rép\_S}} = \frac{TailleDuPaquet}{BandePassanteSortant - débitSortant}$$

Faire correspondre les paramètres de bas niveau aux mesures de haut niveau facilite donc la gestion les ressources par rapport aux spécifications du SLA et prévenir des violations par allocation de ressources supplémentaires. Cette allocation dynamique implique des coûts supplémentaires pour le client, paramètre qui n'a pas été pris en charge par Vincent C. Emeakaroha et son équipe (Emeakaroha, et al. 2010; Emeakaroha, et al. 2012; Maurer, et al. 2012).

2.1.2 Allocation dynamique de ressources

L'idée d'allouer dynamiquement des ressources supplémentaires à une MV afin de prévenir des violations a été prise en partie par LoM2HiS, mais reprise de manière plus complète par Garg B. et coll. (Buyya, et al. 2011; Wu, et al. 2011). Garg B. et son équipe se donnent pour objectif de prévenir des violations du SLA tout en minimisant le coût des ressources allouées.

Les ressources allouées et le coût associé sont validés par le client avant utilisation. Globalement, cette proposition permet de (i) définir les spécifications du SLA, basé sur les paramètres de QdS requis par le client; (ii) Décrire une stratégie de correspondance en interprétant les exigences des requêtes du client en terme de besoin en ressources; et (iii) implémenter un mécanisme d'ordonnancement afin de minimiser les risques de violation en cherchant des MVs ayant les caractéristiques proches de celles qui sont en train d'être utilisées par le client afin de minimiser le coût des allocations. Pour rassurer la correspondance entre les caractéristiques des ressources et les spécifications du SLA, Grag B. et coll. ont utilisé les paramètres suivants :

- ***Type de MV*** : la plupart des fournisseurs font un classement des machines virtuelles en trois types, *large, moyen, petit*; une MV de type *large* a une capacité valant deux MVs de type moyen ou quatre MVs de type *petit*.
- ***Temps d'initiation du service*** : définis le temps nécessaire pour initialiser une MV.
- ***Prix*** : prend en compte combien coûte une MV à un fournisseur pour desservir un client.
- ***Temps de transfert de données*** : le temps nécessaire afin de transférer un Gigaoctet de données d'une MV à une autre.
- ***Vitesse de transfert de données*** : qui dépend du réseau et l'emplacement géographique du client.

Ainsi, lorsque les ressources disponibles pour un client risquent d'être insuffisantes, cette approche propose d'allouer de nouvelles ressources disponibles sur d'autres VMs afin d'éviter de potentielles violations du SLA. Ces VMs sont choisies en fonction de leurs caractéristiques, qui doivent correspondre aux spécifications du SLA qu'a signé le client. En tenant compte des paramètres ci-dessus, les nouvelles allocations ont un faible coût pour le client et apportent des bénéfices au fournisseur qui n'a plus à payer des pénalités.

Cette approche permet de minimiser le coût des allocations pour un fournisseur certes, mais est-ce qu'il peut allouer de nouvelles ressources à tous ses clients? Est-ce que tous les clients ont les mêmes privilèges lorsqu'on sait que la plupart des fournisseurs offrent des services infonuagiques de base gratuits?

Mario Macías et Jordi Guitart (Macias and Guitart 2012a; Macias and Guitart 2012b) ont proposé un modèle d'allocation de ressources en fonction de la priorité du client. Ainsi, ils ont classifié les clients selon un modèle d'affaires qui tient compte du montant facturé à un compte et les privilèges associés en fonction de l'affinité du client avec le fournisseur. L'affinité est utilisée pour classer les clients dans des groupes spécifiques, par ordre de priorité, comme : or, argent, bronze et gratuit. Notant qu'un client classé dans le groupe or est prioritaire à un client classé dans le groupe argent alors qu'un client classé dans celui argent est prioritaire à un client classé dans le groupe bronze. Un client classé dans le groupe bronze est prioritaire à un client ayant accès à un forfait gratuit qui ne paie pratiquement rien pour le service. Ainsi, les clients non prioritaires seront pénalisés afin de satisfaire ceux qui sont prioritaires en cas de surcharge et manque de ressources. Cette approche permet donc de réduire les nombres de violations et augmenter la qualité de service pour les clients prioritaires.

La classification des clients par affinités avec le fournisseur selon un modèle d'entreprise permet de mieux répartir les ressources en fonction de la priorité du client certes. Mais, comment établir une relation de confiance entre le client et le fournisseur? Qu'est-ce qui garantit à un client que tel ou tel fournisseur a la capacité de respecter ses

engagements? Ou encore, comment choisir un fournisseur jouissant d'une bonne réputation quant au respect de ses engagements?

2.2 La confiance

Pour faciliter une relation de confiance entre le client et le fournisseur, nous allons analyser dans cette section certains travaux ayant apporté leur part de contribution.

2.2.1 Confiance : Définition

La littérature regorge de publications traitant le concept de la confiance; certains auteurs sont des références incontournables pour comprendre les différentes nuances de ce concept.

En psychologie, Deutsch Morton (Deutsch 1962) place les notions de coût et bénéfices comme des perceptions individuelles dont dépend la décision de faire confiance. Ainsi, Deutsch montre que dans une décision de confiance, l'individu est confronté à une ambigüité due au fait qu'il y ait deux alternatives : une positive et une négative dont les issues dépendent d'une entité tierce. En choisissant de faire confiance, l'individu donne donc un poids fort à sa perception d'une issue positive en sachant qu'il y a un risque d'une issue négative par rapport à son attente.

En sociologie, Luhmann (Luhmann 2000) place la confiance dans un contexte sociomultidimensionnel, ce qui est différent de Deusch (Deutsch 1962) qui a plutôt placé la confiance dans une dimension individuelle. Pour Luhmann (Luhmann 2000) la confiance permet de réduire la complexité de l'environnement en faisant cas d'un déni élevé de certains facteurs pouvant contrarier notre fonctionnement dans la société. Par exemple, en sortant dans la rue le matin pour vaquer à nos activités, nous acceptons le risque de nous faire agresser, happer par une voiture, impliquer dans un accident, etc. Pour Luhmann, c'est en acceptant de prendre tous ces risques que nous diminuons la complexité sociale. Donc tout comme Deusch (Deutsch 1962), pour Luhmann (Luhmann 2000) la confiance est basée sur la notion du risque.

En sciences sociales et politiques, Gambetta (Gambetta 2000) définit la confiance comme « la probabilité subjective par laquelle un individu prévoit qu'un autre individu exécute une action spécifique dans un contexte dont son bien-être dépend ». Selon Gambetta (Gambetta 2000), la confiance n'est pas généralisée, elle est donc spécifique à un contexte avec une projection vers le futur.

Une des définitions de la confiance la plus proche de notre proposition est donnée dans [20] en se basant sur la définition donnée par Gambetta (Gambetta 2000).

« La confiance qu'une entité C (le client) a dans une autre entité F (fournisseur) pour fournir un Service (S) est la probabilité que cette entité F satisfasse une requête de l'entité C pour le service S. »

De manière générale, les auteurs se mettent d'accord dans leurs définitions de la confiance sur la présence de trois éléments comme décrits dans la figure 6 :

- Une entité, un agent ou personne faisant confiance (*trustor*)
- Une entité, un agent ou une personne cible à qui l'on fait confiance (*trustee)*. Dans un autre sens, c'est le dépositaire de la confiance du *trustor.*
- Un contexte ou une situation dans laquelle la confiance est faite.

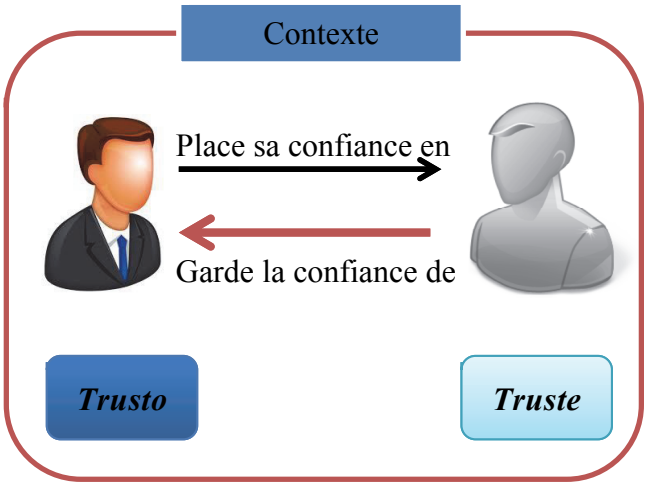


Figure 4 : Les éléments intervenant dans une relation de confiance (Gambetta

2.2.2 Confiance : Généralités

Dans cette section, nous allons présenter les types de confiances, les propriétés, les types de systèmes de confiance afin de mieux apprécier les modèles de confiance proposés dans les systèmes informatiques.

2.2.2.1 Les types de confiance

Dans la littérature, les auteurs font référence généralement à deux types de confiance : la confiance fonctionnelle et la confiance de référence. Josang et Pope (Jøsang and Pope 2005) définissent les deux types de confiance comme suit :

- *La confiance fonctionnelle :* reflète la définition de base de la confiance. C'est le degré de perception de confiance qu'un agent *A* a en un agent *B* à accomplir une tâche.
- *La confiance de référence :* Ce type de confiance implique une référence à une entité tierce. Le degré de confiance qu'une entité *A* place en une entité *B* dépend de la recommandation d'une autre entité *C*.

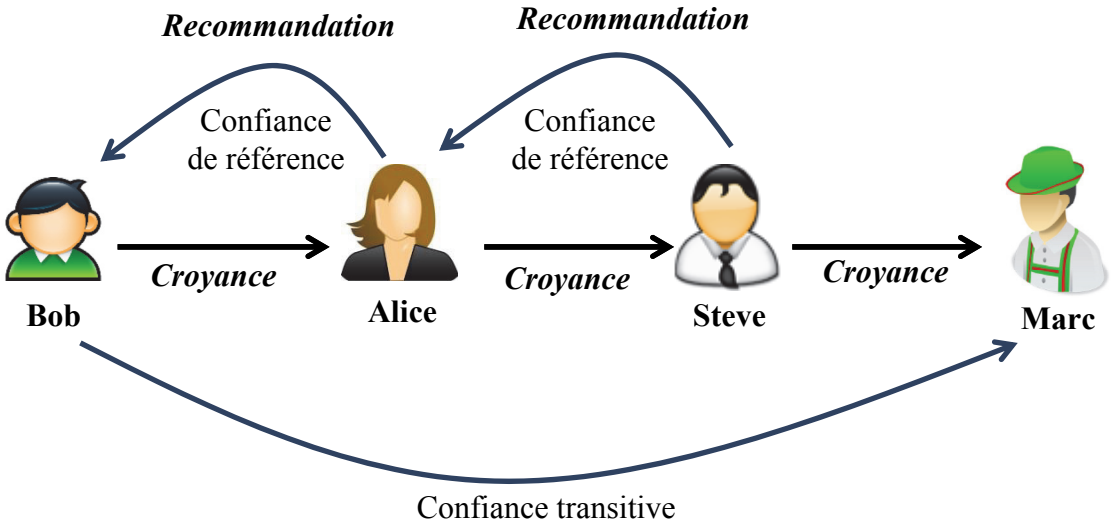


Figure 5 : Les types de confiances (Jøsang and Pope 2005)

2.2.2.2 Confiance : Quelques propriétés

A. Le risque

Plusieurs auteurs placent le risque dans le champ d'application de la confiance comme étant un facteur à ne pas négliger dans une relation de confiance. Paul Marsh (Marsh 1994) mesure le risque encouru dans une relation de confiance comme étant le rapport des coûts et des bénéfices.

$$risque = \frac{Coûts}{bénéfices} \quad (1)$$

Pour Paul Marsh [22], plus le risque est élevé plus l'engagement de coopération est problématique.

B. La symétrie

La confiance n'est pas forcément symétrique. Si une entité A fait confiance à une entité *B*, cela n'implique pas que *B* ait confiance en *A*. Marsh (Marsh 1994) parle de réciprocité, dans le cas où la confiance est symétrique. Pour Marsh (Marsh 1994), la réciprocité diminue le risque et fait augmenter le degré de confiance dans une coopération.

C. L'indépendance

En suivant la définition de Gambetta (Gambetta 2000), l'accent est mis sur une entité. De manière individuelle, chaque entité peut avoir sa propre conception qui définit son degré de confiance dans une autre entité indépendamment des autres.

D. La transitivité

La confiance n'est pas intrinsèquement transitive. Comme le prouve Josang et Pope (Jøsang and Pope 2005) la transitivité est possible que dans certains cas. Par contre, dans le cas de la transitivité, au fur et à mesure que le nombre de références augmente, le niveau de confiance risque de diminuer. Par exemple, si Bob demande la référence d'un dentiste à Alice. Alice répond, ma sœur me parlait d'un dentiste que son ami lui avait recommandé qui lui était référé par un ami de confiance. Le niveau de confiance que Bob va avoir à cette référence est moindre que si s'était un dentiste qu'Alice avait consulté directement.

E. La composition

Les informations recueillies par une entité en provenance d'autres entités de référence, dans un contexte donné, peuvent être assemblées pour donner une perception globale de la décision qu'il faut prendre. Par exemple, Bob demande à ses amis la référence d'un dentiste. Bob va surement bâtir sa confiance autour des références ayant la plus forte fréquence parmi les réponses qu'il aura reçues.

F. La mesure

Pour mesurer la confiance, quatre types de valeurs sont généralement utilisés (Jøsang, et al. 2007; Vu, et al. 2010).

- **Valeur unique** : une valeur unique peut être utilisée pour mesurer la confiance. Cette valeur est mentionnée uniquement si l'entité n'inspire pas confiance et ne fait rien en cas contraire. C'est la mesure utilisée pour assurer la qualité des produits dans une chaine de production par exemple, si un élément ne respecte pas les normes de production on le retire de la chaine et rien n'est signalé en cas contraire.
- **Les valeurs binaires** : les valeurs binaires sont utilisées pour distinguer une entité de confiance ou non. Les valeurs binaires ne permettent pas de distinguer un agent avec qui on a déjà un historique de coopération.
- **Valeurs multiples** : permettent de prendre en compte l'historique de coopération entre deux entités. Par exemple, les valeurs possibles sont des niveaux de confiance « très bas, bas, moyen, haut et très haut ».
- **Les valeurs continues** : les valeurs suscitées sont des valeurs discrètes. Les valeurs continues donnent une gamme plus large de valeurs possibles du niveau de confiance. Généralement, cette valeur varie entre [0,1]; elle mesure la confiance sous forme de probabilité avec un seuil définissant la valeur minimale pour qu'une entité soit considérée comme une entité de confiance.

Pour mesurer le degré de confiance sous la forme d'une valeur quelconque, les systèmes informatiques se sont basés sur des modèles de confiance utilisant des procédés différents selon le besoin. Lesquels nous tâchons de décrire dans la section suivante?

2.2.3 Les modèles de confiance

Les modèles de confiance utilisent une ou plusieurs des propriétés de la section précédente pour calculer, mesurer et évaluer la confiance.

Il existe deux types de modèles de confiance dans la littérature : les modèles à base de politique et les modèles à base de réputation (Jøsang, et al. 2007; Vu, et al. 2010).

2.2.3.1 Les modèles de confiance à base de politique

Les modèles de confiance à base de politique sont aussi appelés modèles de gestion de confiance. Ces modèles utilisent un ensemble de règles qui définissent les conditions à respecter pour être une entité de confiance. Deux entités qui veulent coopérer doivent attendre l'autorisation du modèle qui utilise une valeur binaire autorisant l'interaction en se basant sur les règles de qualifications. Les systèmes utilisant les modèles de gestion de confiance, par exemple, sont les systèmes à clés publique/privée comme PGP[1].

2.2.3.2 Les modèles de confiance à base de réputation

Les systèmes basés sur des modèles de confiance par réputation ne se contentent pas de vérifier si une entité aurait respecté les qualifications requises ou pas, mais ils tiennent compte aussi des interactions et expériences passées avec cette entité (Hussain, et al. 2007; Jøsang, et al. 2007). Par exemple, on ne peut pas affirmer qu'un médecin est de confiance juste en vérifiant son permis de pratiquer la médecine. Il est donc important de tenir compte de ses expériences passées, avec des patients, avant de lancer une telle affirmation.

2.2.4 Les modèles de confiance dans l'infonuage.

Dans l'infonuage, malheureusement, la satisfaction d'une requête du client n'est pas garantie à cent pour cent, ce qui a pour risque de diminuer la perception du niveau de confiance du client à l'égard du fournisseur. C'est pourquoi, avant la négociation du SLA, le client devrait avoir l'option de choisir le fournisseur ayant la capacité de répondre au mieux à ses exigences.

[1] *https://www.gnupg.org/index.html*. Date de la dernière visite 26 juin 2014.

Pour pallier ce problème, M. Firdhous et coll. (Firdhous, et al. 2011) proposent un mécanisme de gestion de confiance qui prend en compte les exigences du client afin de trouver le fournisseur ayant la capacité à mieux satisfaire ce dernier. Ce mécanisme mesure la performance des services en se basant sur le temps de réponse du système. Le score du niveau de confiance d'un fournisseur évolue en fonction de deux critères : (i) les performances mesurées; et (ii) les retours d'expérience (*feedback*) reçus des clients. Mais le temps de réponse à lui seul ne suffit pas pour donner une vision objective du niveau de confiance d'un fournisseur. En effet, un fournisseur peut satisfaire les requêtes de ses clients avec un court temps de réponse pour un service qui n'est pas disponible pour la plupart du temps.

En tenant compte de plusieurs autres paramètres, Chakraborty et Roy (Chakraborty and Roy 2012) proposent un cadre d'applications qui évalue la confiance d'un fournisseur de service Infonuagique en se basant sur des valeurs quantitatives. Cette approche évalue le niveau de confiance d'un fournisseur en deux étapes. La première étape consiste à évaluer les propositions du fournisseur exprimées dans le SLA avant la signature du contrat; les paramètres proposés dans SLA tels que mémoire, processeur, stockage et temps de panne sont extraits et évalués afin de mesurer la capacité du fournisseur à satisfaire les besoins du client. Dans la deuxième étape, des informations sont extraites des fichiers de sessions (p.ex., les temps de réponse, les échecs de connexions dus à la non-disponibilité du système) afin de faire une comparaison entre les promesses faites dans SLA et la QdS réellement fournie. Cette approche, tout comme la précédente ne tient pas compte des interactions historiques du fournisseur afin d'avoir une idée plus ou moins convaincante de sa réputation à respecter ses engagements.

Qiang Guo et coll. (Qiang, et al. 2011) ont développé un algorithme, appelé modèle d'évaluation extensible de la confiance *(ETEC : Extensible Trust Evaluation)*, qui calcule le niveau de confiance d'un fournisseur en se basant sur un système de confiance par recommandations; il fait évoluer le niveau de confiance du fournisseur de manière dynamique en analysant les opérations faites dans le temps. Contrairement aux travaux précédents (Chakraborty and Roy 2012), Qiang et coll. (Qiang, et al. 2011) prennent en

compte les données historiques tout en introduisant un facteur d'oubli afin de ne pas à devoir analyser toutes les interactions d'un fournisseur depuis la mise en ligne du service.

De manière générale, certains de ces travaux comme celui de (Firdhous, et al. 2011) se basent sur le retour d'expérience du client. Cependant, ils sont de plus en plus délaissés au profit d'autres méthodes beaucoup plus automatiques, car en général, le client participe rarement à l'évaluation d'un service. En plus, lorsqu'il le fait ce n'est pas forcément de manière objective ce qui peut biaiser les calculs. D'autre part, la confiance n'est pas uniquement le passé et le présent, c'est aussi une projection vers l'avenir. Dans L'infonuage c'est encore plus qu'évident, car le client signe un contrat dans une perspective d'avenir. Donc, il est tout aussi important de faire des prédictions sur le niveau de confiance d'un fournisseur pour le mois prochain, par exemple, en se basant sur les données historiques.

Si le calcul du niveau de confiance d'un fournisseur est important, le partage et l'interopérabilité des informations sur l'état du système en générale entre les VMs le sont aussi. La façon standard de partager et exploiter ses informations dans le Web se fait par les standards du web sémantique, principalement, par les ontologies.

2.4. Les ontologies dans le web

Sommairement, une ontologie en informatique est une représentation structurée des concepts du modèle d'un domaine quelconque d'un champ d'informations. Elle consiste à représenter les concepts du domaine et les relations existantes entre eux. Elle est donc surtout utilisée pour faire des raisonnements sur le domaine en question. Le champ d'application des ontologies en informatique est très large notamment en intelligence artificielle et surtout dans le domaine médical. Depuis quelques années les ontologies se retrouvent donc en haut des couches déjà standardisées par le consortium W3C pour le web sémantique (Shadbolt, et al. 2006).

2.4.1 Généralité du web sémantique

Le web sémantique[2] est un ensemble de technologies qui fournit un cadre commun visant à rendre le contenu des ressources du web accessible, partageable, réutilisable par des agents logiciels, grâce à un système de métadonnées formelles. L'objectif du web sémantique est d'arriver à un web de données tel qu'il a été imaginé par Tim Berners-Lee, l'inventeur du Web (*World Wide Web)*, qui prévoyait un web de données où le contenu peut être analysé et traité par des machines qui sont capables de communiquer entre elles pour résoudre nos problèmes quotidiens (Berners-Lee and Fischetti 2000). Pour y arriver, le W3C a ajouté un ensemble de couches de technologies au Web actuel, appelé la *Semantic Web Stack* qui est donc l'architecture du web sémantique où chaque couche a sa fonction et entretient des relations avec les couches adjacentes (voir la figure 6).

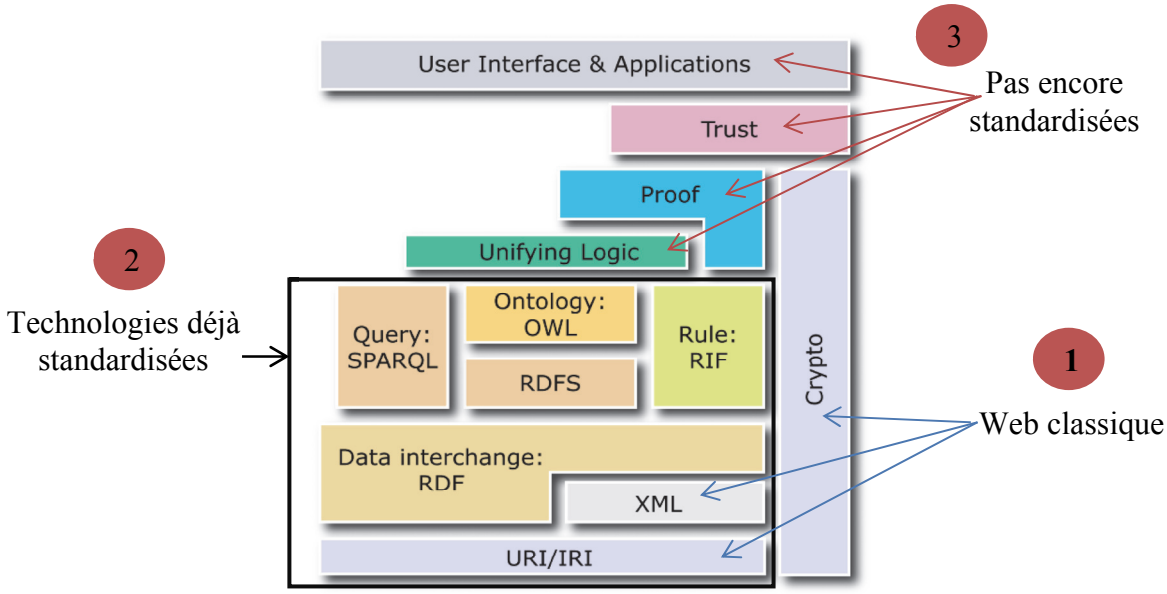


Figure 6 : Pile des standards du web sémantique proposée par W3C[3].

Les couches de technologies déjà standardisées sont :

URI : Identifiant uniforme de ressource *(Uniform Resource Identifier)*. Il représente une référence à une ressource qui n'est pas forcément disponible dans le Web. Un exemple d'URI est donné dans le tableau II.

[2] *http://www.w3.org/2001/sw/* Le Web sémantique est une plateforme coopérative dirigée par le W3C qui travaille sur des standards pour échanger des données dans le Web. Date de la dernière visite 26 juin 2014.
[3] *http://www.w3.org/2006/Talks/0123-sb-W3C-ThingsWeb/#(7)*. Date de la dernière visite le 13 août 2014.

Tableau II : Exemple d'URI

Exemple d'URI
http://mycloudSimulator.com

XML : Langage de balisage extensible *(Extensible Markup Language)*. C'est un langage qui permet de présenter les documents de manière structurée par des balises selon un schéma de données spécifique. Le tableau III donne un exemple de balisage XML.

Tableau III : Exemple de balisage XML

Exemple de balisage XML

```
<actor>
  <provider>Google</provider>
  <administrator/>Jules</administrator>
  <broker>ClouDTrusted</broker>
  <consumers>
      <customer>C-092121</customer>
      <serviceIntegrator>Carlos</serviceIntegrator>
      <endUser>L-73100</endUser>
  </consumers>
</actor>
```

RDF : Resource Description Framework (W3C 2014). C'est un formalisme de description des ressources du web. Les vocabulaires associés à ce formalisme permettent le traitement automatique des données du Web par des applications tierces. Ce langage utilise les données XML pour représenter les relations entre elles sous forme de graphes. Un graphe RDF est représenté par un ensemble d'énoncés; un énonce est un triplet sous forme de (<**S**> <**P**> <**O**>) où S=Sujet (*de quoi on parle*), P=Prédicat (*propriété du sujet*) et O=Objet (*Valeur de la propriété de S*). Dans un graphe RDF, les nœuds représentent les ressources et les arcs décrivent les relations entre les ressources. Un exemple de triplet RDF, tiré du RDF 1.1 Primer (W3C 2014) du W3C, est fourni dans le tableau IV.

Tableau IV : Exemple de triplets RDF (W3C 2014)

Exemple de triplets RDF

```
<Bob> <est une> <personne>.
<Bob> <est ami avec> <Alice>.
<Bob> <est né le> <4 juillet 1990>.
<Bob> <est intéressé par> <la Mona Lisa>.
<la Mona Lisa> <a été créée par> <Leonardo da Vinci>.
<la vidéo "La Joconde à Washington"> <est à propos de> <la
Mona Lisa>
```

Le graphe RDF correspondant aux triplets du tableau IV est présenté dans la figure 7.

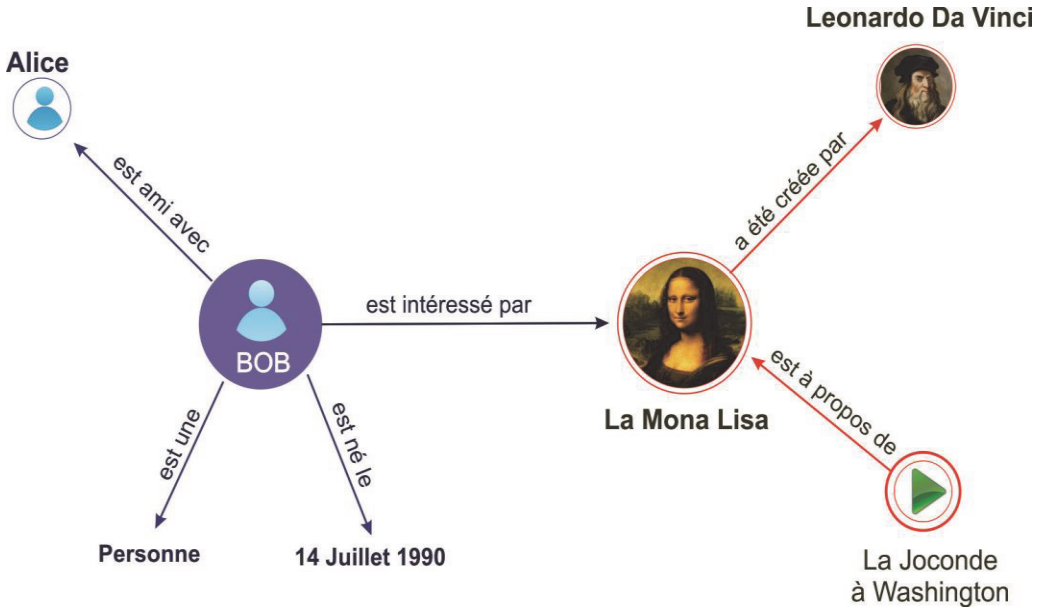


Figure 7 : Exemple de graphe RDF avec des triplets (W3C 2014)

RDFS : *RDF Schéma*. C'est un langage taxonomique de représentation de connaissances dans le Web permettant de structurer les ressources RDF. Il permet de décrire les relations entre les ressources tout en définissant la hiérarchie qui existe entre elles. Par exemple (voir tableau V), RDFS permet d'exprimer des énoncés ensemblistes par exemple : « Tout Auteur est une personne »

Tableau V : Exemple d'expression RDFS

Exemple d'expression RDFS

```
<rdfs:Class rdf:ID= " Personne "/>
<rdfs:Class rdf:ID= "Auteur ">
  <rdfs:subClassOf rdf:resource="#Personne"/>
</rdfs:Class>
```

SPARQL : *SPARQL Protocol and RDF Query Language*. Ce langage permet d'avoir accès aux données RDF. En somme, SPARQL est pour le RDF ce que SQL est pour les bases de données.

Les ontologies OWL : OWL (Langage d'ontologie pour le Web) (Motik, et al. 2009) est un langage descriptif qui, en se basant sur les données RDF et des règles, permet de faire des raisonnements sur un domaine de connaissance. Mettre en place une ontologie avec OWL requiert :

- Identification des classes ou entités et définition de leur hiérarchie. Par exemple, les classes Fournisseur et Client sont des sous-classes de la classe Acteur.
- Identification et définition des classes disjointes.
- La définition des propriétés des classes et leur hiérarchie. Par exemple, « un Client a payé le service d'un Fournisseur.». *Payer* est donc une propriété de la classe Client.

OWL utilise une base de connaissances et des règles pour faire des inférences à partir du langage des logiques descriptives (LD). Cette base de connaissances est composée de deux compartiments (Figure 10) : (1) Tbox : est utilisée pour la définition des concepts (classes) et leurs propriétés (p.ex., Animal, Fruits, Livre); (2) Abox : là sont déclarées les instances ou assertions, des individus correspondant aux concepts (p.ex., Chat, Pomme, Le petit livre rouge).

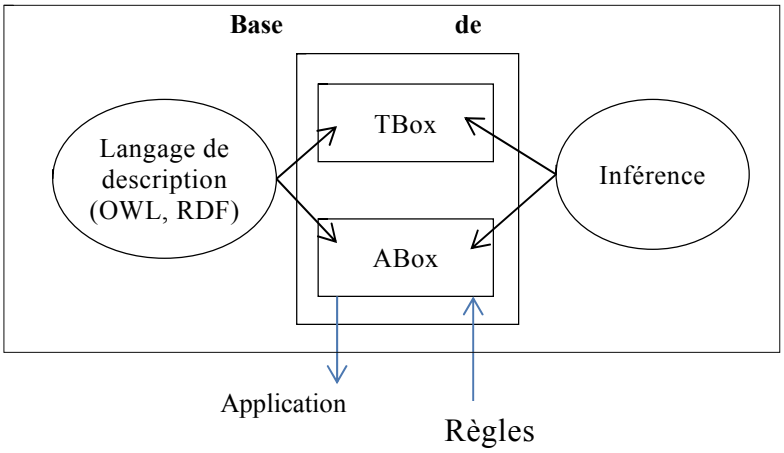


Figure 8 : Base de connaissance OWL — LD

RIF : *Rules Interchange Format*. Afin de donner une capacité de raisonnement plus large aux ontologies, RIF ajoute un formalisme de règles permettant aux ontologies d'inter-changer et de partager des règles. Au-delà de la classification des concepts et leurs relations, OWL utilise les langages de règle pour faire des inférences. Plusieurs moteurs d'inférences sont basés sur ce formalisme qui facilite l'interopérabilité et l'interprétation automatique des données comme dans le projet *Linking Open Data* [4] dont un exemple est le DBpedia[5].

2.4.2 Les ontologies dans l'infonuage

Échanger les données entre les infrastructures infonuagiques dans un format standard et léger facilitera l'interopérabilité des services. Mais aussi, l'exploitation de ces données, par des ontologies pouvant faire des inférences en se basant sur des règles, peut faciliter la gestion des ressources, anticiper les violations et assurer le respect du contrat.

Assurer l'interopérabilité, entre les différents formats de SLA produits par différents fournisseurs de service infonuagique, et évaluer le niveau de réputation de chacun de ces fournisseurs par une entité tierce, appelée centre de qualité, sont deux principaux objectifs

[4] http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData. Date dernière visite 14 août 2014
[5] http://dbpedia.org/About. Date dernière visite 14 août 2014.

de Liu et coll. (Liu, et al. 2012). Ils font appel à OWL pour créer une ontologie basée sur la sémantique des SLA sous un format unique, XML, facilitant leur interopérabilité. Comme dans les travaux précédents (Emeakaroha, et al. 2010; Emeakaroha, et al. 2012; Maurer, et al. 2012), (Buyya, et al. 2011; Wu, et al. 2011), (Liu, et al. 2012). l'équipe de Liu (Liu, et al. 2012) extrait des informations du SLA afin de faire des évaluations sur la QdS offerte et calculer le niveau de réputation du fournisseur en analysant les données historiques pour déterminer les violations par inférence tout au long du cycle de vie du contrat

Si Liu (Liu, et al. 2012) a mis l'accent sur la réputation du fournisseur, Yong Beom et coll. (Yong Beom, et al. 2011) par contre, ont penché sur la gestion des ressources et la répartition des tâches afin d'éviter des violations du SLA. Yong Beom (Yong Beom, et al. 2011) décrit les machines virtuelles comme des entités avec des propriétés et qui sont liées par des relations bien définies avec objectif de satisfaire les spécifications du SLA; pour représenter toutes ces relations, Yong Beom (Yong Beom, et al. 2011) utilise une ontologie OWL/RDF basée sur un ensemble de règles permet d'allouer automatiquement de nouvelles tâches à une MV candidate sélectionnée par inférence.

Certains travaux ont déjà porté leur part de contribution dans la mise en place d'une ontologie pour la qualité de service dans les services web en général, c'est le cas, entre autres, de QoSOnt (Dobson, et al. 2005) , WS-QoSOnto (Vuong Xuan 2008), DAML-QoS (Chen, et al. 2004). En se basant sur ces travaux, Kaouthar et coll. (Dobson, et al. 2005) ont développé SLAOnt qui consiste à analyser un ensemble de modèles de SLA, exprimer les obligations de ces contrats sous forme de règles avec le langage SWRL (*Semantic Web Rule Language*) pour prévenir des violations et mettre en place une ontologie autour construite avec OWL. SLAOnt (Fakhfakh, et al. 2008) comprend une interface de gestion permettant de déterminer une violation en fonction des règles inférées et la gestion automatique des SLAs.

Une limitation majeure de ces travaux (Dobson, et al. 2005) , (Vuong Xuan 2008), (Fakhfakh, et al. 2008) entre autres, c'est qu'ils sont basés sur une ontologie comme OWL qui est basée sur une logique binaire, descriptive et déterministe. En effet, OWL peut déterminer une violation, mais ne peut pas en prédire en utilisant la base des données analysées. OWL peut aboutir à une conclusion par inférence; cependant, elle ne permet pas

de quantifier le pourcentage de validité ou non de cette conclusion en tenant compte des incertitudes qu'on risque d'avoir autour de cet énoncé. Ainsi, dans ce travail, nous allons utiliser une ontologie probabiliste, compatible avec OWL, capable de tenir compte de l'incertitude autour d'une conclusion. Considérons, par exemple, un scénario décrit par OWL au tableau VI.

Tableau VI : Scénario décrit par OWL

Scénario décrit par OWL
<Bob> <est le mari de > <Alice>
<Alice et Bob> <ont eu> <deux enfants>
<Jacques> <est le fils de > <Bob>
<Marise> <est la fille de > <Alice>

Dans ce scénario, on aura tendance à conclure par inférence que Jacques et Marise sont frères et sœurs. Cependant, si on utilise un langage probabiliste, pour décrire ce scénario, on pourra donner la probabilité qu'une telle conclusion soit vraie ou fausse en tenant compte de l'incertitude que Jacques et Marise soient nés d'une autre relation antérieure ou extraconjugale. Dans les services web en général et dans l'infonuage en particulier, avec un modèle de consommation sans limites « *Pay as You Go !* », les incertitudes sont grandes sur le comportement du client qui peut faire un nombre de requêtes variables d'un moment à l'autre; donc, l'utilisation d'une ontologie probabiliste s'avère nécessaire afin d'avoir des conclusions plus pertinentes.

2.4.3 Les ontologies probabilistes

À date, il n'y a pas encore d'ontologie probabiliste qui soit standardisée par le W3C. Par contre, certains travaux (p.ex., BayesOWL (Ding, et al. 2006) et PR-OW[6]) ont

[6] http://www.pr-owl.org/ *A Bayesian extension to the OWL Ontology Language*.2010. Date de la dernière visite 14 juin 2014.

proposé des solutions assez pertinentes; le travail le plus documenté, qui a déjà un noyau de projet largement utilisé, est PR-OWL (Carvalho, et al. 2013; Costa, et al. 2008) qui est déjà à sa version 2.0 compatible avec OWL

P. Costa (Costa 2005), un des premiers instigateurs du projet PR-OWL, définit une ontologie probabiliste comme étant une représentation formelle et explicite de la connaissance concernant un domaine d'application. Autre que les propriétés d'OWL décrites à la sous-section 2.4.1, une ontologie probabiliste comprend :

- Des régularités statistiques qui caractérisent le domaine;
- Une définition pour des connaissances incomplètes, peu fiables, ambiguës et peu concluantes
- L'incertitude concernant les connaissances citées ci-haut.

2.4.4 L'ontologie probabiliste PR-OWL

PR-OWL est une extension d'OWL, compatible avec la version 2.0 d'OWL, ayant la capacité de faire des raisonnements probabilistes basés sur les réseaux bayésiens avec entités multiples (MEBN : *Multi Entity Bayesian Network*). MEBN est un langage probabiliste de premier ordre, il représente les concepts du monde comme des entités ayant des relations entre elles. PR-OWL fait donc une représentation formelle et explicite des connaissances du domaine étudié. Outre les différentes caractéristiques d'OWL, PR-OWL tient compte des régularités statistiques des champs et conclut sur de nouvelles formes de connaissance par inférence tout en tenant compte de l'incertitude autour de ces connaissances (Costa, et al. 2008).

PR-OWL utilise une autre forme de combinaison de classes, au lieu de (<sujet><prédicat><objet>) d'OWL, et les relations qui les relient ensemble. La figure 9 présente les classes dans un ovale et les relations qu'elles entretiennent ensemble sont indiquées par des flèches.

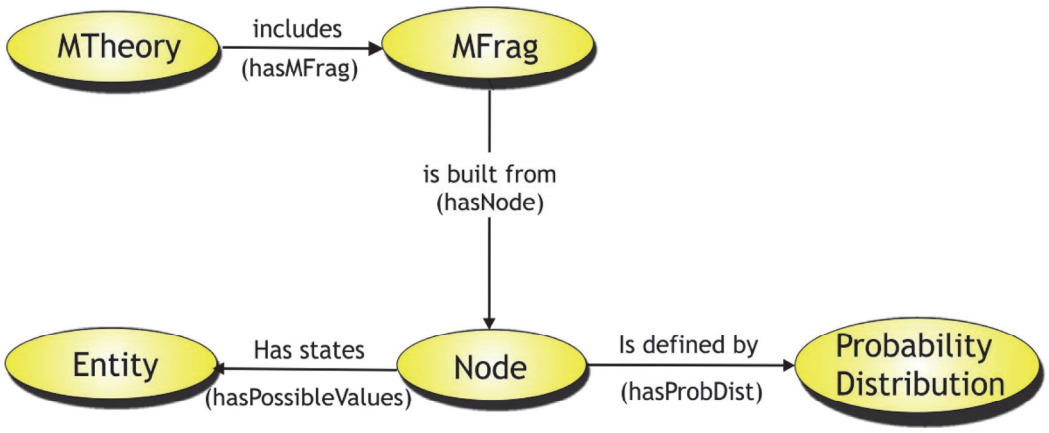


Figure 9 : composition d'un MTheory de PR-OWL (Costa 2005)

De manière générale, la classe MTheory représente un concept du monde, pour lequel l'ontologie va être construite. Un MTheory est un regroupement de MFrag qui correspond à des sous-classes ou des propriétés de classes destinées à acquérir de nouvelles connaissances. Un MFrag regroupe les variables aléatoires (Node) d'une classe sont représentées sous forme de nœuds résidents (*Resident node*) et de contexte (Contexte Node) où le MTheory en est le réseau. Chaque nœud a des états qui lui sont propres appelé des entités (Entity); les nœuds sont définis par une table de distribution des probabilités conditionnelles (TPC) qui définit la connaissance apriori du modèle. Les MFrags calculent la probabilité jointe de chacune de ses variables aléatoires tant dis que PR-OWL utilise les informations encapsulées dans les Mfrags pour répondre aux requêtes probabilistes (Laskey, et al. 2011)..

Les principales différences entre OWL et une ontologie probabiliste comme PR-OWL se résument dans le tableau VII suivant.

Tableau VII : Principale différence entre OWL et PR-OWL

OWL	PR-OWL
Standardisé	*Non standardisé*
Langage déterministe	*Langage probabiliste*
Ne tient pas compte de l'incertain	*Tient compte de l'incertain*
Base logique	*Réseau bayésien*
Compatible : XML, RDF	*Compatible : XML, RDF*
Formalisme : Sujet, prédicat, objet	*MTherory, MFrag*

2.5 Conclusion

Le SLA est le contrat qui régit le service infonuagique entre le fournisseur et le client. Ce contrat est souvent violé par le fournisseur qui en conséquence, subit des pertes économiques et risque de souffrir d'une mauvaise réputation en offrant une mauvaise qualité de service à ses clients. Dans ce chapitre, nous avons présenté et analysé des travaux de recherche réalisés pour résoudre ces problèmes en visant une application efficace du SLA dans l'environnement de l'infonuage. Ces travaux sont présentés selon trois angles différents qui sont : la gestion des ressources, la gestion de confiance, les ontologies (*Le tableau IX présente une comparaison, entre les travaux analysés dans ce chapitre, tout en montrant leurs limites*). Nous avons constaté que la gestion des ressources joue un rôle particulier dans la plupart de ces travaux. Ce rôle consiste à extraire des informations des paramètres de bas niveau pour des mesures de haut niveau afin de prévenir des violations par des calculs comparatifs ou par des ontologies. Nous avons constaté que les ontologies utilisées, étant basées sur OWL, sont bâties sur des langages déterministes. Elles ne permettent pas de prédire et anticiper les cas de violations tout en tenant compte des incertitudes autour des données recueillies. Nous avons constaté aussi que les méthodes utilisées pour calculer le niveau de confiance d'un fournisseur tiennent compte des données historiques uniquement pour évaluer la réputation de ce dernier sans une projection vers l'avenir afin de prédire la tendance du niveau de confiance du fournisseur en question. D'autres méthodes évaluent le niveau de confiance du fournisseur en se basant sur un retour d'expérience du client qui n'est pas forcément objectif dans ses

évaluations. En constatant la faiblesse de ces travaux (voir tableau IX) en termes de prédictions, anticipations et les incertitudes, nous avons proposé un cadre d'applications, détaillé au chapitre 3, qui permet à un client de choisir un fournisseur de confiance tout en projetant cette confiance dans un futur proche. Nous avons aussi proposé une ontologie probabiliste qui permet au fournisseur d'anticiper les cas de violation. Notre approche étant basée sur des réseaux bayésiens, elle tient compte de l'incertitude pouvant exister dans les données fournies en entrées.

Tableau VIII : Comparaison et limitation des travaux *(X : n'a pas été pris en considération)*

Contribution	Gestion des ressources	Gestion de confiance	Ontologie
LoM2HiS (Emeakaroha, et al. 2010; Emeakaroha, et al. 2012; Maurer, et al. 2012)	Correspondance des paramètres de bas niveau des ressources en mesures de haut niveau des paramètres du SLA.	X	X
Garg et coll. (Buyya, et al. 2011; Wu, et al. 2011)	Allocation dynamique des ressources. Analyse de la QdS.	X	X
Mario Macías et Jordi Guitart (Macias and Guitart 2012a; Macias and Guitart 2012b)	Classification des clients selon un modèle d'entreprise et allocation des ressources selon la priorité de ces derniers.	X	X
M. Firdhous et coll. (Firdhous, et al. 2011)	X	Retour d'expérience du client et la capacité du fournisseur à satisfaire les exigences du client.	X
Chakraborty et Roy (Chakraborty and Roy 2012)	X	Analyse des données du fichier log.	X
Qiang Guo et coll. (Qiang, et al. 2011)	X	Réputation par recommandation après analyse des données historiques.	X
Liu et coll. (Liu, et al. 2012)	Assure le respect de la QdS.	Analyse des données historiques et calcule la réputation du fournisseur selon le nombre de violations trouvées.	OWL
Yong Beom et coll. (Yong Beom, et al. 2011)	Relations et propriétés entre les MV.	X	OWL, allocation de tâches par inférence à une MV candidate.
SLAOnt (Fakhfakh, et al. 2008)	X	X	OWL, SWRL. Détermination les violations et gestion automatisée des SLAs.

Chapitre 3 : Conception du cadre d'applications

Prévenir les violations du SLA dans l'environnement de l'infonuage est un défi auquel plusieurs travaux se sont déjà penchés comme on vient de le voir dans le chapitre précédent. Dans le cadre de cette recherche, nous avons mis l'accent sur deux questions principales : comment un client peut-il choisir un fournisseur de confiance ayant une bonne réputation? Et, comment prédire des violations en analysant les données historiques?

Pour répondre à ces deux questions, nous avons proposé un nouveau cadre d'applications comprenant trois modules que nous pouvons décrire brièvement de la manière suivante :

1. Le module de confiance : peut être géré par une entité tierce. Ce module est responsable de prédire le comportement d'un fournisseur en analysant les données historiques de ce dernier. Ensuite, il envoie les résultats dans un répertoire qui est responsable de classer les fournisseurs selon leur réputation.
2. Le module intelligence : est basé sur une ontologie probabiliste capable de tenir compte des incertitudes autour des données de sessions recueillies.
3. Le module de contrôle : est l'interface qui classifie les clients, et intercepte le cas de violations prévues par le module intelligence.

3.1 Architecture générale du cadre d'applications

Cette section décrit l'architecture non détaillée du cadre d'applications, illustrée dans les figures 10 et 11, dans un premier temps. Dans un second temps, elle décrit les outils et concepts utilisés pour la mise en place d'un tel cadre d'applications.

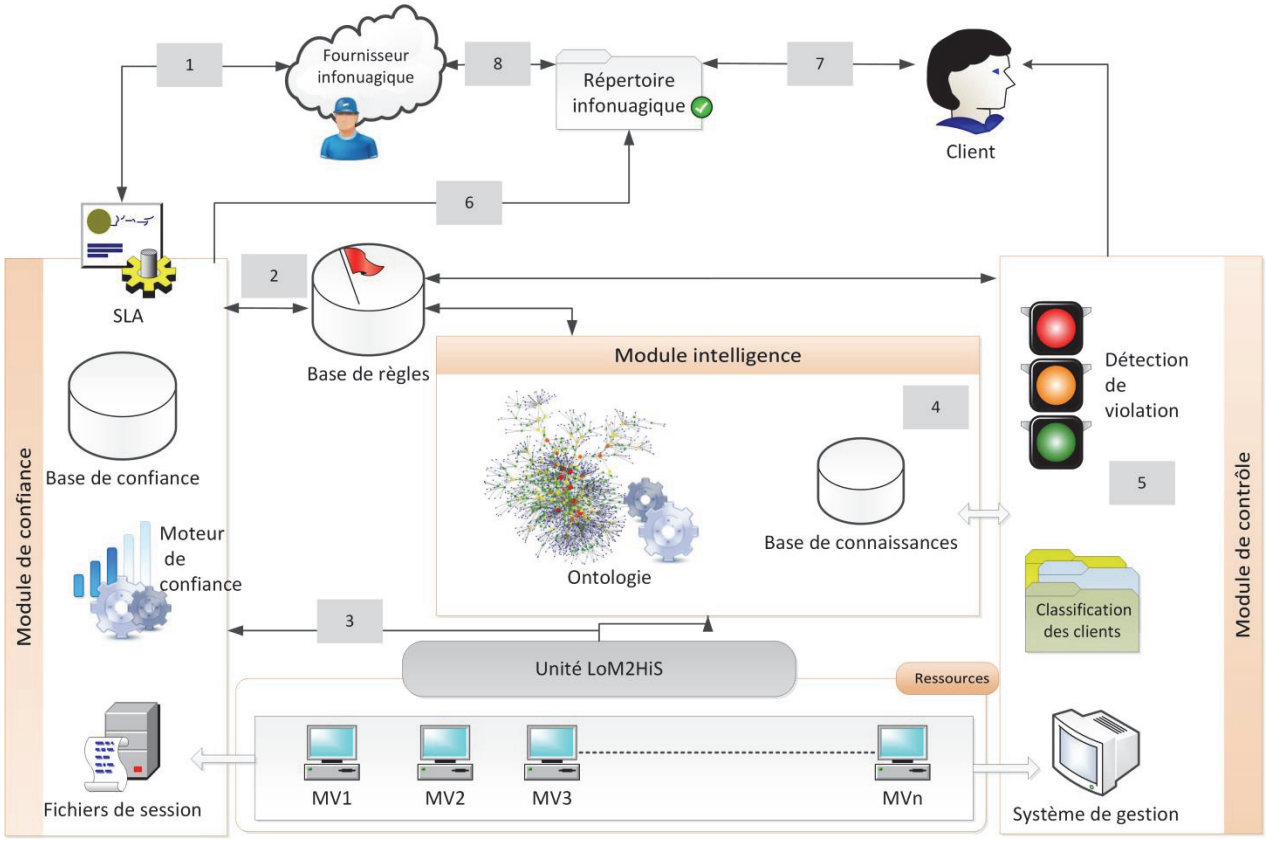


Figure 10 : Architecture globale du cadre d'application

L'architecture globale du cadre d'applications illustrée sur la figure 10 présente les relations existantes entre les entités suivantes :

- Le fournisseur décrit les modalités, qui sont les politiques et les règles du service dans le SLA (1), qui sont extraites, partagées entre les modules et sauvegardées dans la base des règles (2).
- L'unité LoM2HiS fait la correspondance entre les paramètres de bas niveau des ressources en paramètres de haut niveau du SLA (3). La correspondance des paramètres est soumise au module de contrôle qui vérifie la disponibilité des ressources par rapport aux spécifications règlementaires du SLA.
- Le module intelligence anticipe les violations à travers le moteur d'inférence de l'ontologie probabiliste (4), il renvoie les résultats au module de contrôle qui affiche les prédictions des violations (5) et en informe le client de nouveaux forfaits adaptés à sa consommation. Les historiques des violations sont sauvegardés dans la base de connaissances.
- Le module de contrôle affiche les cas de violation et classifie les clients selon la politique du fournisseur. Les informations de classification des clients sont envoyées au module d'intelligence qui les récupère au niveau de l'ontologie.

- Le module de confiance extrait les informations historiques des fichiers de sessions en provenance des MVs pour faire des prédictions à travers le moteur de confiance. Les niveaux de confiances calculées par le moteur de confiance sont sauvegardés dans la base de confiance et affichés dans l'interface du répertoire infonuagique (6).
- Un client voulant avoir un service infonuagique doit accéder au répertoire (Répertoire *infonuage*) (7) contenant la liste des fournisseurs avec leur score de niveau de confiance calculé par le module de confiance, afin de choisir le fournisseur désiré avant d'entamer la négociation du contrat avec ce dernier (8).

L'ordonnancement de la communication entre les entités du cadre d'applications est décrit sur le diagramme de séquence illustrée dans la figure 11.

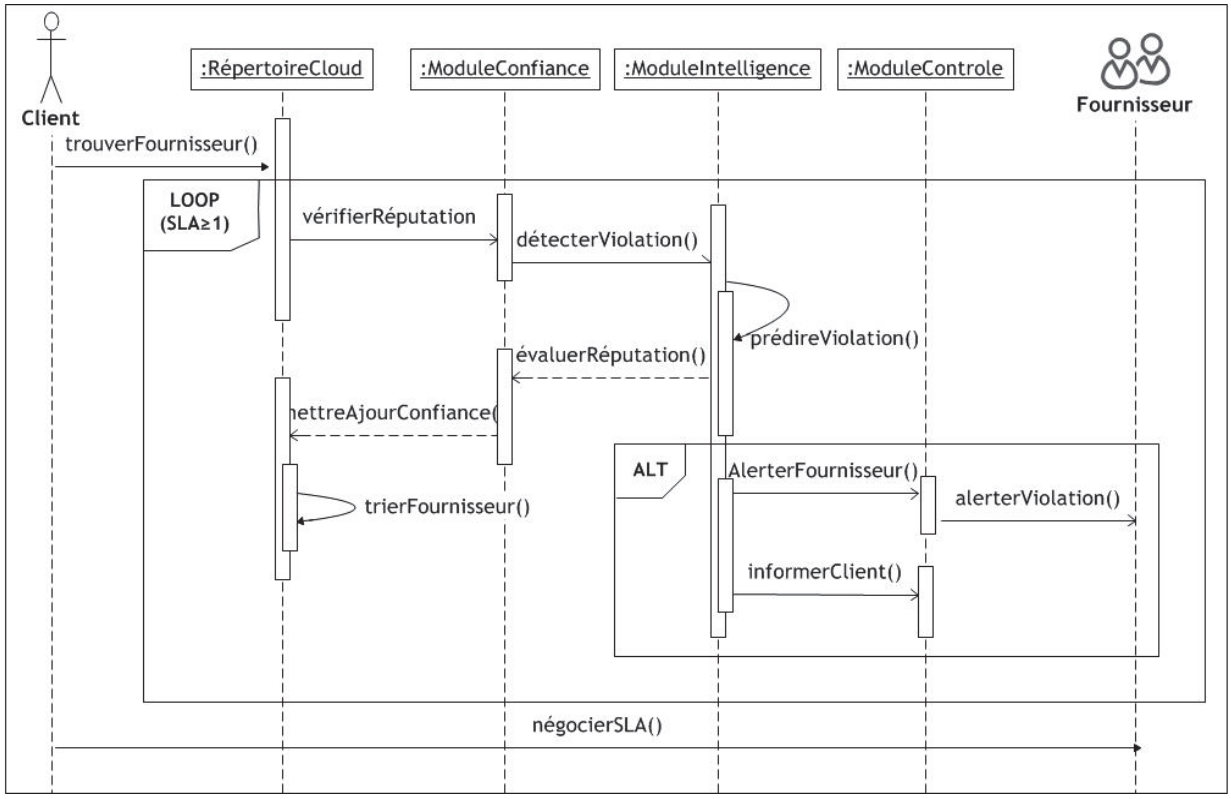


Figure 11 : Interaction entre les modules du cadre d'applications

3.1 Architecture détaillée du cadre d'applications

3.1.1 Le module de confiance

Pour faire des prédictions sur le niveau de confiance d'un fournisseur en analysant les données historiques de sa réputation on a utilisé un réseau bayésien. Le score de confiance d'un fournisseur est la probabilité a posteriori de sa réputation calculée par inférence bayésienne. Les réseaux bayésiens, pour faire des prédictions, sont largement utilisés dans le domaine médical notamment pour faire des diagnostiques tant qu'en intelligence artificielle (Cooper and Herskovits 1992; d'Ambrosio 1991; Heckerman 1997).

Le calcul des probabilités dans un réseau bayésien suit la règle de Bayes. Étant donné un ensemble de données historiques d'un service infonuagique (D), la règle de Bayes permet de calculer la probabilité a posteriori qu'il ait une violation, notée hypothèse (H).

$$p(H \mid D) = \frac{p(D \mid H).p(H)}{p(D)} \tag{2}$$

Où :

P (H|D) : est la probabilité a posteriori de (H) sachant les données (D)

P (H) : est la probabilité a priori de (H)

P (D|H) : est la probabilité de vraisemblance de (H) sachant (D)

P (D) : est la probabilité d'observation de D peu importe que (H) soit vrai ou faux.

Par exemple, soit **D** un ensemble de données historique contenant des cas de violations. Nous voulons prédire la probabilité que les violations viennent d'une mauvaise disponibilité du service notée **H**. L'équation (2) permet de poser la probabilité apostériori p (H|D) où p (D|H) est la probabilité d'observation de H dans D. P (H) est la probabilité apriori de H, cette valeur peut être donnée par un expert ou omise, dans la première itération. Par contre, la valeur de la probabilité apostériori de l'itération précédente peut devenir la probabilité apriori de l'itération en cours. P (D) est la probabilité d'observer des cas de violations en présence de mauvaises disponibilités ou non.

Dans notre cas, les données historiques peuvent être acquises en deux étapes :

1. La première étape consiste à extraire les paramètres du SLA qui décrivent la QdS que le client veut recevoir du fournisseur (p.ex., disponibilité, temps de réponse, stockage, et stockage) et faire la liaison entre eux et les paramètres de bas niveau avec LoM2HiS.
2. La deuxième étape consiste à extraire les données des fichiers de sessions et faire des comparaisons entre les promesses du SLA dans l'étape 1 afin de noter les cas de violation.

Ces données historiques sont introduites dans un réseau bayésien pour pouvoir prédire par inférence le comportement du fournisseur en question en lui accordant un score de confiance.

La figure 12 illustre un réseau bayésien naïf utilisé pour mettre en place notre modèle de confiance.

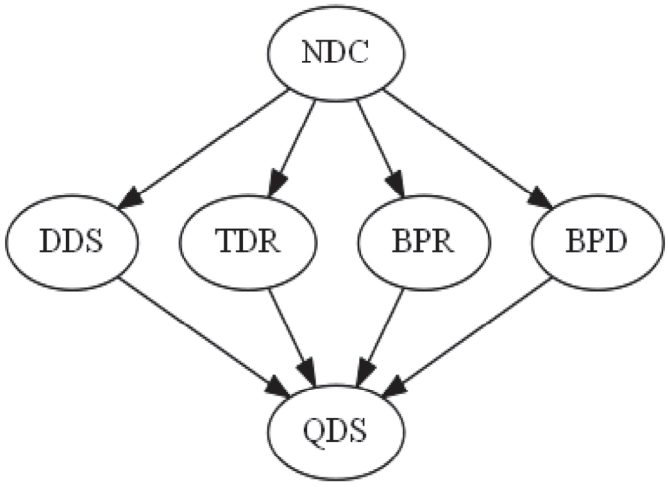


Figure 12 : Réseau bayésien naïf du module de confiance

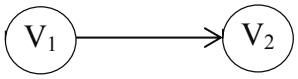
Tableau IX : Description des variables du réseau bayésien

Variable	Description
NDC	Niveau de confiance
DDS	Disponibilité du service
TDR	Temps de réponse
BPR	Bande passante du réseau
BPD	Bande passante disque
QDS	Qualité de service

3.1.1 Généralités des réseaux bayésiens

Formellement, un réseau bayésien est défini comme un graphe orienté acyclique noté **G = (V, E)** où **V** est l'ensemble des variables (nœuds/sommets) aléatoires et **E** l'ensemble des arcs représentant les liens causaux entre les variables. Ce réseau est aussi caractérisé par des probabilités conditionnelles finies dans l'espace **(Ω, Z, P)**. Où **P** est une loi de probabilité appliquée à l'événement (**Ω, Z)** dans l'univers **Ω**.

De manière intuitive, pour représenter un lien de cause à effet à l'aide d'un graphe une flèche est généralement utilisée de la façon suivante :



Implicitement, cela signifie que, la connaissance que nous avons sur $\mathbf{V_1}$ influence notre connaissance sur $\mathbf{V_2}$, souvent noté $\mathbf{V_1}$ implique $\mathbf{V_2}$. Cette référence est réciproque dans les réseaux bayésiens une fois que la loi de probabilité de $\mathbf{V_2}$ est connue, $\mathbf{V_1}$ peut être donc déduit en observant $\mathbf{V_2}$.

Ceci permet d'exprimer la loi jointe des probabilités en fonction de la structure du réseau de la manière suivante :

$$p(V_1,.....,V_n) = \sum_{i=1}^{n} p(V_i \mid C(V_i)) \qquad (3)$$

*Où **C (Vi)** l'ensemble des causes ou parents de Vi dans **G***

3.1.1.1 L'inférence bayésienne

L'inférence bayésienne consiste à calculer la probabilité conditionnelle d'un ou plusieurs nœuds du réseau à partir d'une base de connaissances donnée apriori (Heckerman 1997). Plusieurs méthodes proposées permettent de calculer l'inférence en fonction des variables de nature discrète (Lauritzen and Spiegelhalter 1988), continue (Shachter and Kenley 1989) ou dans un réseau mixte (Lauritzen 1992).

3.1.1.2 Apprentissage bayésien

Faire des prédictions avec les réseaux bayésiens consiste à calculer la probabilité par inférence sur toute la propriété du graphe. Le calcul de ces probabilités sur un ensemble de données exige la connaissance (1) des paramètres, c'est-à-dire, la probabilité conditionnelle à l'intérieur des nœuds; et (2) de la structure du réseau qui représente au mieux les probabilités observées. Ces deux connaissances sont acquises par l'apprentissage bayésien (Naïm, et al. 2011)

3.1.1.3 Apprentissage des paramètres

La probabilité d'observation d'un état quelconque parmi les états possibles à l'intérieur d'une variable du réseau pour l'ensemble des données est un paramètre. L'apprentissage des paramètres consiste à estimer la distribution des paramètres pour l'ensemble des données selon la loi jointe des probabilités. En gardant comme hypothèse que la structure du réseau est connue, deux méthodes sont généralement utilisées pour l'apprentissage des paramètres : l'approche statistique et l'approche bayésienne.

3.1.1.4 Approche statistique

L'apprentissage statistique des paramètres est appliqué dans le cas d'une structure connue avec des données complètes. Pour un ensemble de paramètres notés ($\theta_1 \ldots \theta_n$), l'approche statistique consiste à calculer la fréquence d'observation de chaque occurrence de l'évènement θ_i dans l'ensemble des données D. Cette fréquence est appelée Maximum de vraisemblance *(ML, maximum Likeliwood*).

$$\hat{P}(X_i = x_k \mid pa(X_i) = x_j) = \hat{\theta}^{ML}_{i,j,k} = \frac{N_{i,j,k}}{\sum_k N_{i,j,k}} \tag{4}$$

Où $\mathbf{N_{i,\ j,k}}$ est le nombre d'évènements dans l'ensemble des données pour chaque fois la variable $\mathbf{X_i}$ est dans l'état $\mathbf{x_k}$ tandis que ses parents sont dans la configuration $\mathbf{x_j}$. Dans la pratique il risque d'avoir de paramètres manquants dans la base de données, il est donc

impossible d'appliquer le Maximum de vraisemblance. Dans ce cas, l'algorithme EM (Espérance-Maximisation) (Dempster, et al. 1977 b) est appliqué afin de chercher les paramètres jusqu'à la convergence optimale de l'espérance et la maximisation. L'espérance estime les $\mathbf{N_{i,\,j,k,}}$ manquants en calculant la moyenne conditionnelle du réseau. La maximisation remplace les $\mathbf{N_{i,\,j,k,}}$ par leur moyenne calculée par l'espérance.

A. Approche bayésienne

L'approche de l'estimation bayésienne consiste à trouver les paramètres θ_i dans les données observées en appliquant une valeur a priori (*prior)* sur les paramètres. La règle de Bayes permet d'écrire

$$\begin{aligned} Posterior &= Likelihood\ .Prior \\ p(\theta|D) &= p(D|\theta).p(\theta) = L(D|\theta).p(\theta) \end{aligned} \tag{5}$$

La prédiction bayésienne utilise le MAP (Maximum a posteriori) afin de tenir compte de l'incertitude lors des prédictions sur l'ensemble du réseau. Ce qui revient à écrire :

$$\hat{P}(X_i = x_k \mid pa(X_i) = x_j) = \hat{\theta}^{MAP}_{i,j,k} = \frac{N_{i,j,k} + \alpha_{i,j,k} - 1}{\sum_k (N_{i,j,k} + \alpha_{i,j,k} - 1)} \tag{6}$$

Où $\boldsymbol{\alpha_{i,\,j,k} - 1}$ sont les distributions de Dirichelet associées aux aprioris (*prior*) sur les paramètres.

Dans le cas où les données sont incomplètes, l'approche bayésienne applique l'algorithme EM de Dempster(Dempster, et al. 1977a) pour apprendre les paramètres.

$$\theta^{(t+1)}_{i,j,k} = \frac{N^*_{i,j,k} + \alpha_{i,j,k}}{\sum_k (N^*_{i,j,k} + \alpha_{i,j,k})} \tag{7}$$

Initialisation du prior de Dirichelet

$X_v = \{X_v^{(l)}\} l = 1...N$

Les paramètres du réseau

$\theta^{(t)} = \{\theta_{i,j,k}^{(t)}\}$

Répéter:

Espérancce

$$N_{i,j,k}^{*} = E[N_{i,j,k}] = \sum_{i=1}^{n} p(X_i = x_k \mid pa(X_i) = x_j, X_v^l, \theta^{(t)}$$

Maximisation

$$\theta_{i,j,k}^{(t+1)} = \frac{N_{i,j,k}^{*}}{\sum_k N_{i,j,k}^{*}}$$

Tant que:

$|\theta^{(t)} - \theta^{(t-1)}| \geq \varepsilon$

Figure 13 : Algorithme EM de Dempster

Cet algorithme fait une recherche vers une convergence optimum locale des variables manquantes. L'algorithme EM est efficace et applicable tant dans une approche statistique que bayésienne.

B. Apprentissage de la structure

Après avoir trouvé les paramètres du réseau, l'apprentissage de la structure permet de trouver la meilleure structure du réseau qui correspond au mieux à la meilleure solution de prédiction. L'apprentissage de la structure consiste à maximiser par randomisation le score (***s)*** de chaque sous-structure et combine leurs résultats afin de trouver le meilleur modèle pour l'ensemble des données. Le nombre des différentes structures pour un réseau bayésien de (***n)*** variables est défini par la formule :

$$r(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-i)} = n^{2\partial(n)} \quad (8)$$

Le score ***S*** de l'ensemble du réseau **(B)** est une composition de la somme des scores ***(s)*** des sous-structures en fonction de chaque sommet et de ses nœuds parents **(pa)**. Il est défini comme suit :

$$S(B) = \sum_{i=1}^{n} s(V_i, pa(V_i)) \quad (9)$$

*Où **n** est le nombre de sommets dans le graphe.*

Cette façon de procéder consiste à faire une recherche exhaustive dans tout le réseau. Ceci augmente exponentiellement le nombre de sous structures possibles, empêchant ainsi son application dans un réseau ayant plus de 8 nœuds. Pour résoudre ce problème, le score bayésien de Dirichelet peut être appliqué pour trouver le meilleur score :

$$scoreDB = p(\beta) \int_{\theta} L(D \mid \theta, \beta) p(\theta, \beta) d\theta \quad (10)$$

Cooper et Herskovits (Cooper and Herskovits 1992) proposent l'algorithme K2 qui limite la recherche dans l'espace du réseau en maximisant la probabilité des sous réseau. Le théorème de Cooper et Herskovits pour l'algorithme K2 stipule pour une base de données **D** ayant pour **N** le nombre de paramètres, avec ***B*** la structure du réseau. En posant $\mathbf{Pa_{ij}}$ la $\mathbf{j^{ième}}$ observation de **Pa** $\mathbf{(X_i)}$ et $\mathbf{N_{ijk}}$ le nombre de cas dans **D** où $\mathbf{X_i}$ vaut $\mathbf{x_{ik}}$. Soit $\mathrm{Nij} = \sum_{k=1}^{r_i} N_{ijk}$ alors l'algorithme K2 implémente l'équation :

$$P(\beta, D) = P(\beta) P(\prod_{i=1}^{n} \prod_{j=1}^{qi} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (11)$$

En supposant que la distribution des paramètres dans la structure du réseau est uniforme, on peut préciser le nombre maximum de nœuds parents pour chaque variable noté (MaxPa). De ce fait, au lieu de faire une recherche dans les **n** sous réseaux, l'heuristique proposée par (Cooper and Herskovits 1992) va permettre de sélectionner un jeu de parents

pour chaque variable. Alors l'équation de sélection d'un jeu de parent optimal pour un nœud $\mathbf{X_i}$ s'écrit :

$$score(X_{i,} pa(X_i)) = \prod_{j=1}^{qi} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}\,!) \qquad (12)$$

Pour *i = 1* **à** *n* **faire**
pa (Xi) = 0;
S = *score (Xi, pa (Xi))*
Tant que suite **et** pa (Xi)| < *MaxPa* ***faire***
Chercher Z précédant Xi qui maximise score (Xi, pa (Xi) ∪ {Z})
Nouveau= score (Xi, pa (Xi) ∪ {Z})
Si *nouveau* > *S* **alors**
S=nouveau
pa (Xi) = *pa (Xi) ∪ {Z})*
Sinon suite = **FAUX**
Fin Tant que
Fin Pour

Figure 14 : Algorithme K2 de Cooper et Herskovits (Cooper and Herskovits 1992)

Cet algorithme ne fait pas une recherche gloutonne dans l'apprentissage de la structure, mais il maximise la recherche dans les sous-structures du réseau en choisissant les variables parentes optimales pour chaque variable courante.

3.2.2 Le module intelligence

De même que le module de confiance, l'ontologie probabiliste PR-OWL est aussi basée sur les réseaux bayésiens.

Ce module comporte :

- Une base de connaissances des violations et des prédictions précédentes.
- Un moteur d'inférence basé sur les réseaux bayésiens à entités multiples (MEBN, *Multi-Entity Bayesian Network*).

3.2.3 Le module de contrôle

Spécifiquement basé sur l'unité LoM2HiS, ce module s'occupe essentiellement à alerter sur l'état du système et les violations détectées. Il communique essentiellement avec le module intelligence dans le but de prévenir le fournisseur des risques de violation et l'état général des ressources.

3.3 Conclusion

Dans le cadre de cette recherche, nous avons implémenté et testé deux modules du cadre d'applications : le module de confiance et le module d'intelligence dont les méthodes d'évaluation et d'implémentation sont décrites dans le chapitre 4. En analysant les interfaces de contrôle développées dans certains travaux passées en revue au chapitre 2, entre autres (Emeakaroha, et al. 2010; Emeakaroha, et al. 2012; Maurer, et al. 2012) (Macias and Guitart 2012a; Macias and Guitart 2012b) , nous avons jugé de donner priorité aux méthodes utilisées :

- Dans la gestion des ressources qui ne tenaient pas compte des incertitudes
- Dans la gestion de confiance qui ne permettait pas de faire des prédictions
- Dans la gestion des violations qui ne permettaient pas l'anticipation.

Le développement du module de contrôle sera considéré dans la suite de notre travail dans un futur proche.

Chapitre 4 : Implémentation et expérimentation

Dans ce chapitre nous allons présenter les méthodes d'implémentations et l'expérimentation de notre proposition ainsi que les outils utilisés à sa réalisation.

4.1 Implémentation

L'implémentation de notre cadre d'applications se fait en deux phases. La première phase consiste à implémenter le modèle de confiance et le tester avec un échantillon de données extraites des sites web des fournisseurs de services infonuagiques tels qu'Amazon EC2, Google CE. La deuxième phase concerne l'implémentation du module intelligence en utilisant une ontologie probabiliste.

4.1.1 Implémentation du module de confiance

Pour mettre en place notre module de confiance nous avons utilisé BNT[7](*Bayesian Net Toolbox*) un outil disponible sous licence libre. Il dispose un ensemble de bibliothèques et de fonctions facilitant l'implémentation et l'évaluation d'un réseau bayésien pour MATLAB (*Matrix Laboratory*) (MathWorks 2013) (MathWorks). L'implémentation du module de confiance comprend les étapes suivantes :

1. Création du réseau ***trust_bnet*** en utilisant la structure initiale présentée dans la figure 14 avec les états possibles de chaque nœud. Dans notre cas les nœuds peuvent avoir deux états : (*1 =bas et 2=haut*). La création du réseau dans BNT se fait par la fonction ***mk_bnet()*** qui prend en paramètre la structure du réseau, la nature des variables (discrètes, continue, mixte), les variables observées.
2. Utilisation d'un moteur d'inférence bayésien (par exemple, arbre de conjonction) en appelant la fonction ***jtree_inf_engine(trust_bnet)*** implémentée dans BNT sous

[7] https://code.google.com/p/bnt/ (BNT) Bayesian Network Toolbox extension disponible en ligne pour MATLAB. Dernière visite le 16 août 2014.

MATLAB (MathWorks 2013) (MathWorks) dans la et identification des variables à observer.

3. Calcul des probabilités conditionnelles de chaque variable à partir d'une base de connaissances avec l'apprentissage des paramètres. En partant de l'hypothèse que la base de connaissances peut avoir des données manquantes, nous avons appliqué l'algorithme EM (voir la figure 13) sur le réseau pour l'apprentissage des paramètres. L'algorithme EM est implémenté par la fonction ***learn_params_em();*** dans BNT. Elle prend en paramètre la base de données de test, la base de données d'entrainement et le nombre d'itérations maximales prévues.
4. Apprentissage de la structure qui optimise au mieux les paramètres appris en appliquant l'algorithme de score K2 (Cooper and Herskovits 1992) (Robinson 1977). L'algorithme K2 est implémenté par la fonction BNT ***learn_struct_K2 ();*** qui prend en paramètre l'ensemble des données, la structure du réseau initial et le nombre de parents maximal qu'une variable peut avoir dans le réseau.
5. Calcul des probabilités marginales avec le moteur d'inférence qui donne les prédictions avec leur vraisemblance. La fonction ***marginal_nodes();*** prend en paramètres le moteur d'inférence et l'évidence de certaine observation sur les variables fin de calculer les probabilités marginales.

4.1.2 Implémentation de l'ontologie probabiliste basé sur PR-OWL

Nous avons utilisé UnBBayes,[8] un outil disponible sous licence libre, écrite en java pour implémenter l'ontologie probabiliste. Cet outil offre une interface graphique et intègre dans ses composants une interface de Protegé(Research Protegé ontology editor and knowledge acquisition system). Une ontologie OWL2 construite avec Protegé est interopérable avec UnBBayes en format OWL/XML. UnBBayes supporte le format PR-OWL2 afin de construire des ontologies probabilistes en utilisant les réseaux bayésiens avec entités multiples pour faire des inférences sur l'ontologie.

[8] http://sourceforge.net/projects/unbbayes/. Date dernière visite 26 juillet 2014.

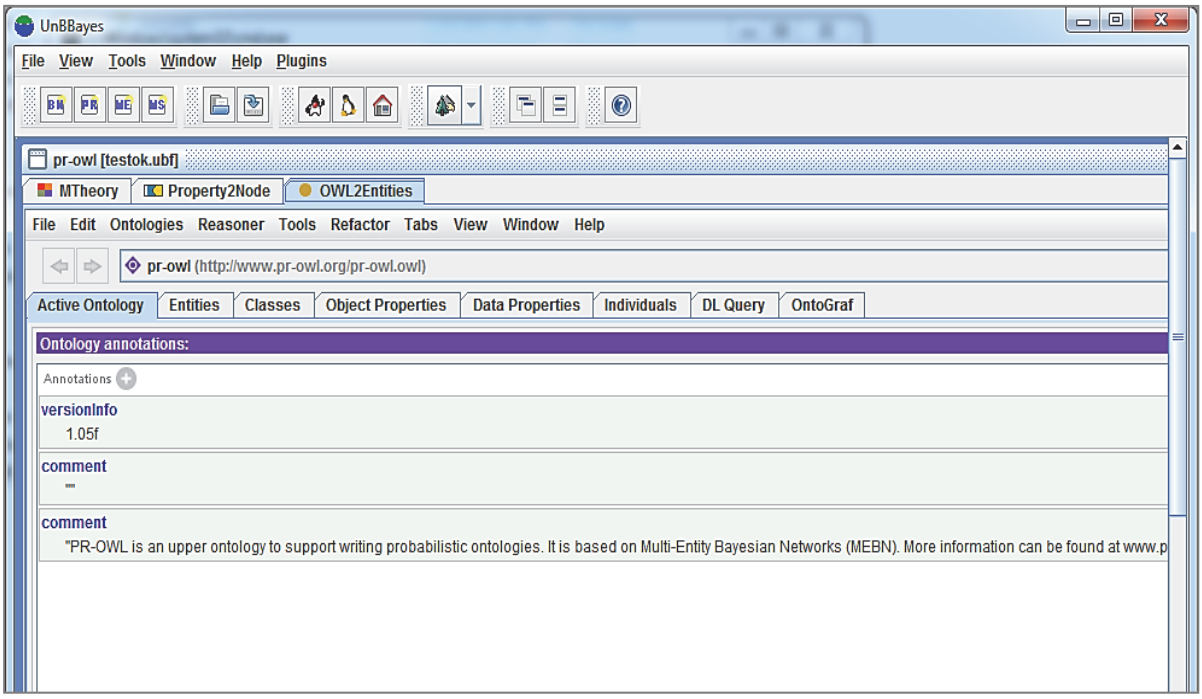


Figure 15 : Interface Protegé dans une fenêtre Unbbayes

Nous avons utilisé les caractéristiques du SLA pour construire l'ontologie. En effet, UnBBayes permet de construire une ontologie entière dans l'interface de Protegé avec ses classes, ses propriétés et ses règles et de les faire correspondre avec leur MFrags approprié le tout regroupé en MTheory. Ainsi, notre ontologie contient des classes comme SLO regroupant deux sous-classes comme propriétés quantitatives (*QuantitativeProperty)* et propriétés qualitatives. La classe propriété quantitative contient des sous-classes comme temps de réponse (ResponseTime), les paramètres des MVs, données (*DataPerformance),* performance réseau (*NetworkPerformance*) disponibilité du service (*Availability*), etc..

Les caractéristiques liées aux politiques de crédit et des prix sont récupérées dans les classes respectives (CreditPolicy) et (Price).

La classe Acteurs (*Actors*) qui regroupe, comme son nom l'indique, les différents acteurs de l'infonuage comme Fournisseur (*Provider*), Courtier (*Broker*), Consommateurs (*Consumers*) regroupant les clients (*Customer*), les utilisateurs finaux (*EndUser*), etc… .

La gestion des ressources disponibles est capturée par la classe Contrôle (*Monitoring*) en terme de Rapport (*Report*), gestion des ressources *(ResourceControl*), détection de violation (*ViolationDetection*).

Nous avons utilisé certaines propriétés associées aux classes par exemple (*isAvailable*) pour déterminer la disponibilité d'une MV, ou (*hasResponseTimeReport*) pour capturer le rapport du temps de réponse de cette dernière.

Pour refléter la réalité et construire des règles appropriées, nous avons implémenté un programme en C# qui extrait les informations des SLA de la page web Amazon EC2[9] et de la page SLA de Google CE[10]. Ensuite, une extraction sur le prix de certains services a été faite sur les pages web respectives de Amazon EC2[11] et de Google CE[12]. Le tableau X qui suit donne une idée des résultats des extractions.

Tableau X : Informations extraites des SLA et prix des services de Amazon EC2 et Google CE

	Amazon EC2			**Google CE**			
Politique de crédit du SLA	***Disponibilité***		***Pénalité***	***Disponibilité***			***Pénalité***
	99.95 % et plus			99,95 % et plus			
	Entre 99.95 % et 99.0 %		10 %	Entre 99.00 % et 99.95 %			10 %
	Moins de 99.0 %		30 %	Entre 95.00 % et 99.00 %			25 %
				Moins de 95.00 %			50 %
Services et Prix	**Type**	**Mémoire (GB)**	**Prix/h.**	**Type**	**Processeur**	**Mémoire**	**Prix/h.**
	m3.large	7,5	0.14 $	n1std-4	4	26 GB	0.28 $
	r3.xlarge	30,5	0.35 $	Highcpu4	4	3.60 GB	0.18 $
	i2.xlarge	30,5	0.85 $	g1-small	1	1.70 GB	0.85 $

[9] *« Amazon Elastic Cloud Compute SLA. »* Disponible à partir *:* http://aws.amazon.com/ec2-sla/. Dernière visite le 17 août 2014.

[10] *« Google Compute Engine SLA ».* Disponible à partir *:* https://developers.google.com/compute/sla?hl=es. *Dernière visite le 17 août 2014.*

[11] *« Amazon EC2 Pricing* ». Disponible à partir : http://aws.amazon.com/ec2/pricing/. *Dernière visite le 17 août 2014.*

[12] *« Google. Compute Engine. Pricing »;* Disponible à partir *:* https://cloud.google.com/products/compute-engine/. Dernière visite le 17 août 2014.

Les politiques de crédit d'Amazon EC2 et Google CE diffèrent sur certains critères, mais les deux offrent une disponibilité du service supérieur à 99.95 %. Pour une disponibilité située entre 99.00 % à 99.95 %, les deux prévoient un remboursement de 10 % du prix. Par contre pour une disponibilité inférieure à 99.00 % Amazon EC2 se consent à rembourser 30 % du prix. Tandis que Google CE rembourse 25 % pour une disponibilité située entre 95.00 % et 99.9 %, et 50 % pour une disponibilité inférieure à 95 %.

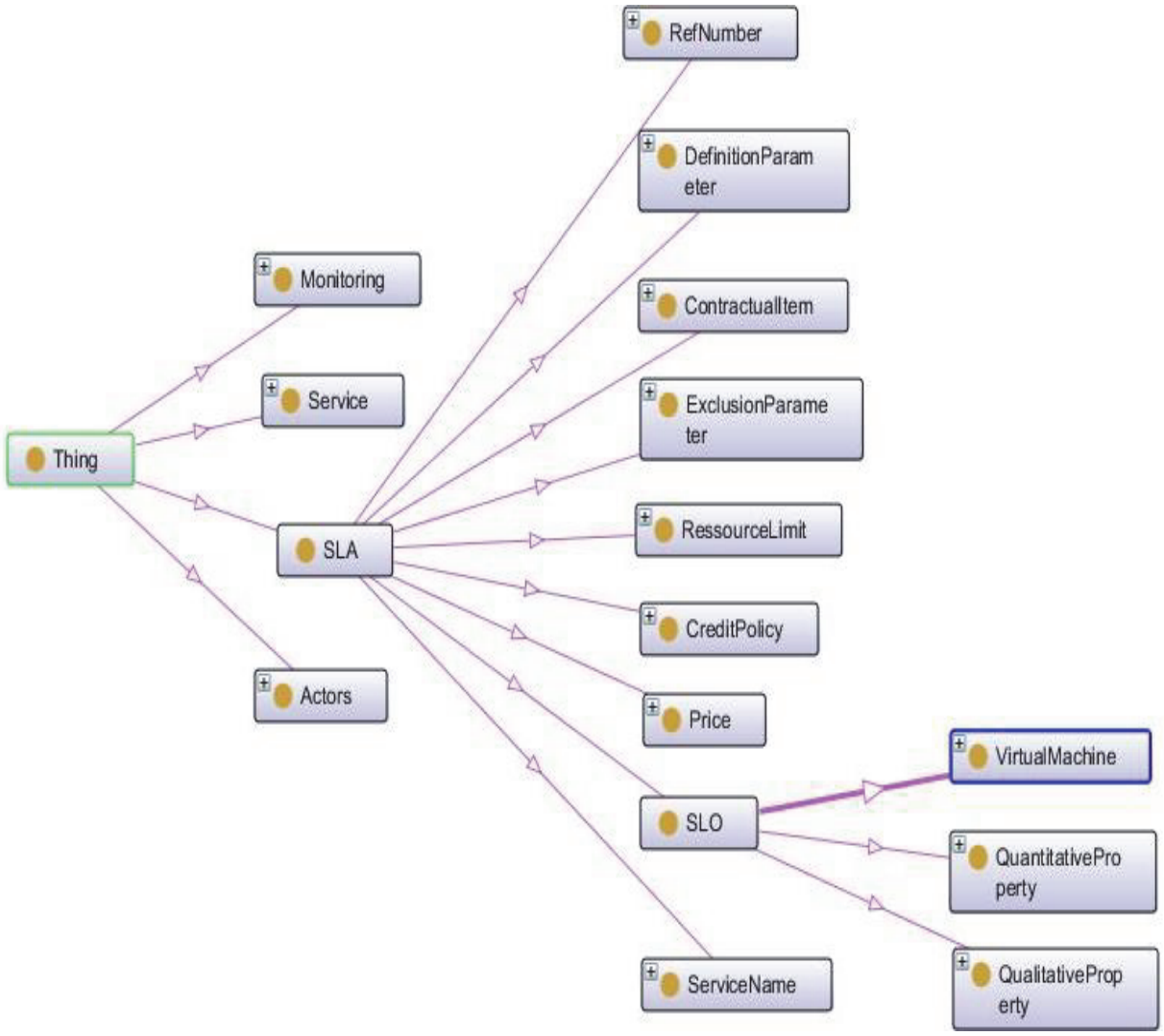


Figure 16 : Vue partielle des principales classes de l'ontologie PR-OWL

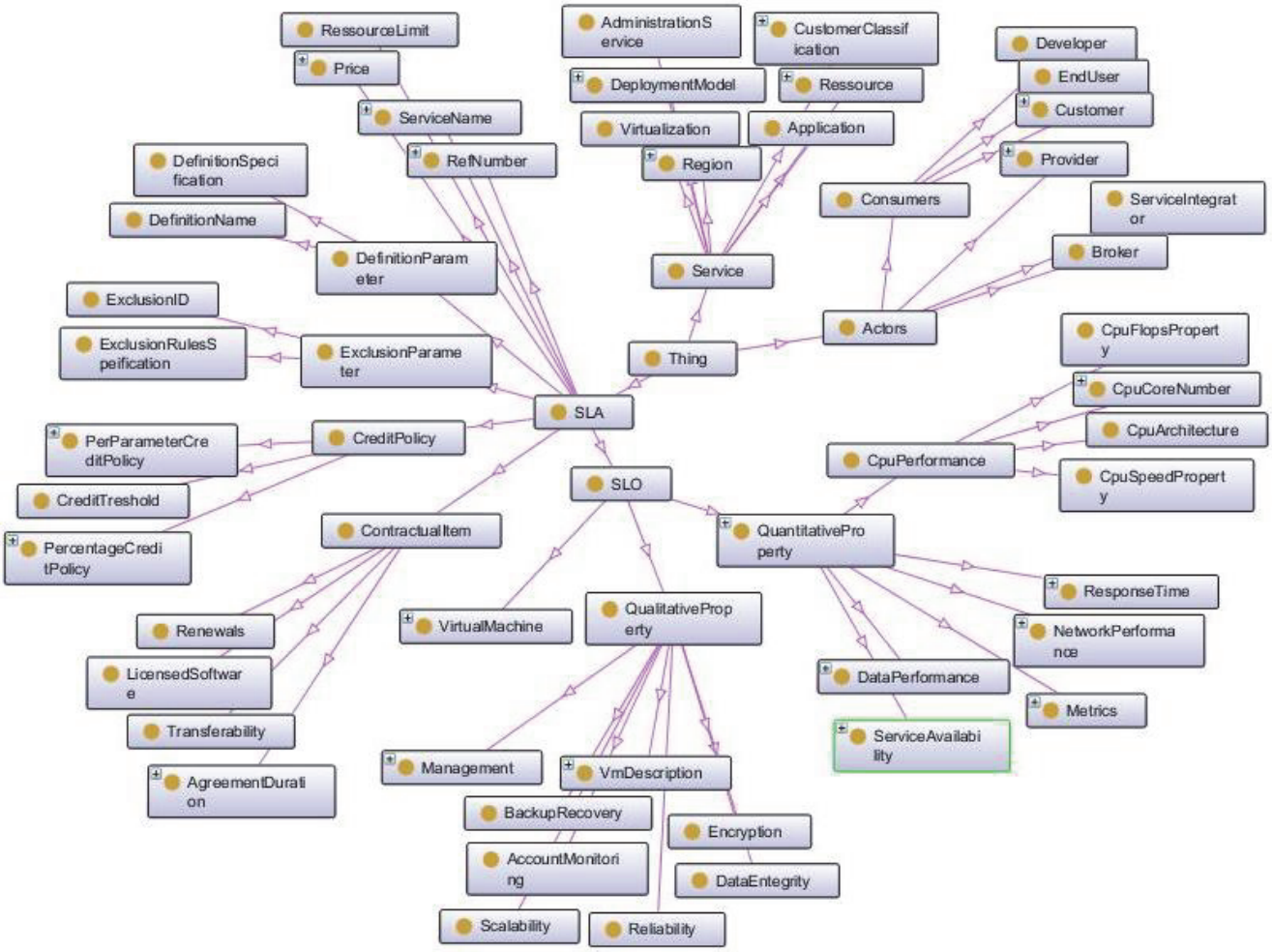


Figure 17 : Vue complète des classes et sous classes PR-OWL

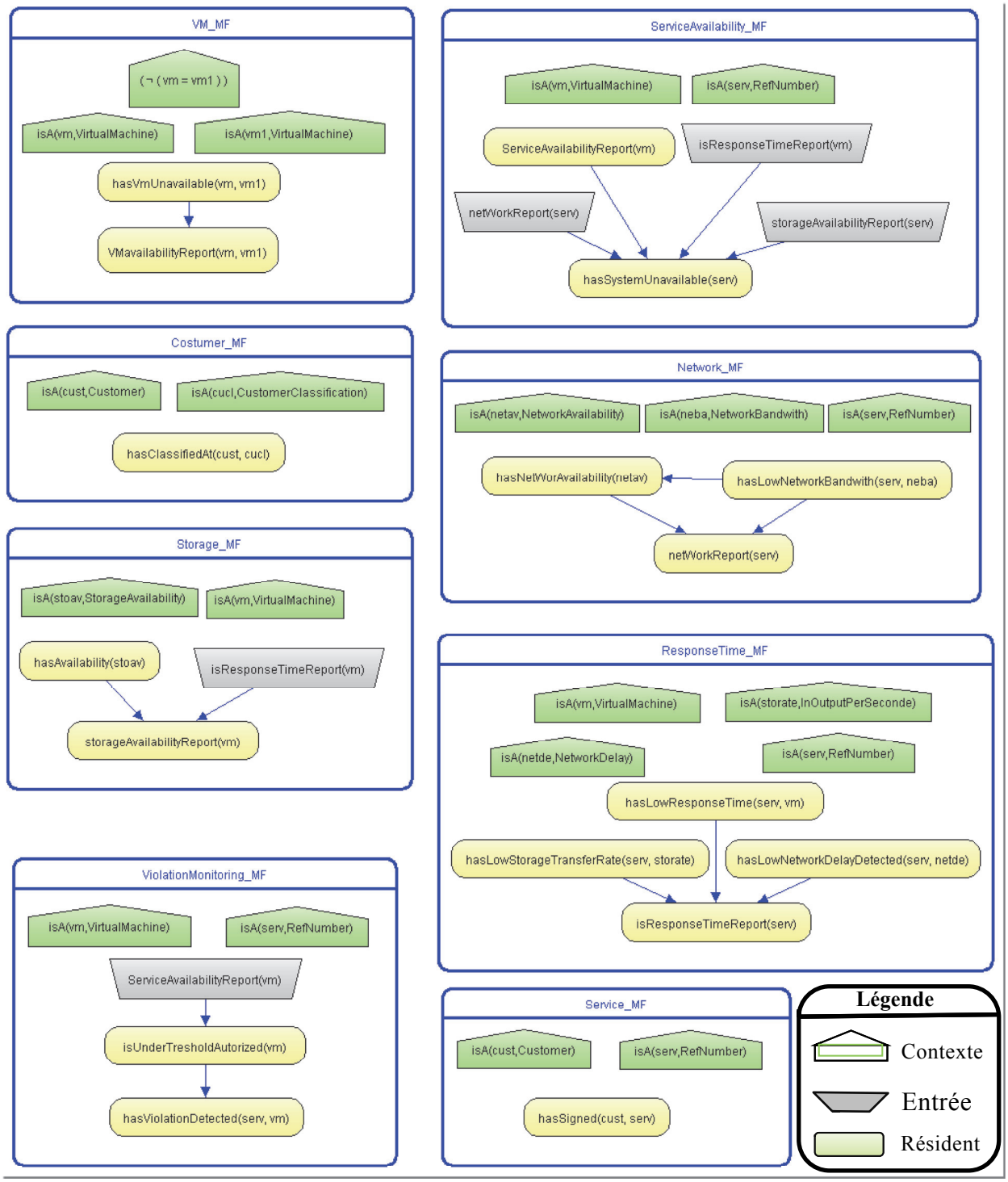


Figure 18 : MTheory de l'ontologie SLA vue par Unbbayes

La figure 18 illustre le MTheory du SLA correspondant en partie à l'ontologie présentée aux figures 16 et 17. Le MFrags comme ***VM_MF*** par exemple, traite les connaissances sur les machines virtuelles. Nous avons ajouté des MVs notées (***VM_1, …VM_n***) comme des entités de la classe ***VM.*** Pour deux MVs distinctes *(¬ (Vm=Vm1))* ***VM_MF*** vérifie la disponibilité de ces MVs par le nœud résident correspondant à la propriété *hasVmUavailable* et *VMavailabilityRepport* se charge de retourner le rapport de la disponibilité du service. Ce scénario est presque identique à tous les MFrags sauf ceux qui attendent une valeur en entrée comme c'est le cas du MFrag ***ViolationMonitoring_MF***. La détection de violation de la disponibilité d'une MV, entre les n VM disponibles, se fait en vérifiant la disponibilité de celle-ci par rapport au seuil prédéfini, 99.95 % dans notre cas, par le nœud résident *isUnderAutorizedTreshold*.

4.2 Évaluation

Cette section décrit l'évaluation du cadre d'application de notre proposition, particulièrement le module de confiance et le moteur d'inférence du module intelligence. Cette évaluation a pour objectif de vérifier et tester la faisabilité de notre proposition qui consiste à donner un score de confiance à un fournisseur en se basant sur les données historiques et prévenir des violations du SLA par des prédictions à partir d'une ontologie probabiliste.

4.2.1 Évaluation du module de confiance

Les étapes de l'évaluation du module de confiance sont les suivantes :

4.2.1.1 Création de la structure du réseau

Nous avons gardé la structure initialement présentée dans la figure 12 avec les variables dans l'ordre décrit dans le tableau XI suivant

Tableau XI : Création de la structure du réseau

Variable ou nœud	Description	Numéro
NDC	Niveau de confiance	1
DDS	Disponibilité du service	2
TDR	Temps de réponse	3
BPR	Bande passante du réseau	4
BPD	Bande passante disque	5
QDS	Qualité de service	6

4.2.1.2 Échantillonnage des données de test

Nous avons créé une base de données d'entrainement contenant 750 enregistrements par un processus aléatoire avec MATLAB en nous basant sur le rapport de M. Gagnaire et coll. décrit dans (Gagnaire, et al. 2012). Ce rapport (Gagnaire, et al. 2012) présente un condensé des temps d'arrêt des services pour 13 fournisseurs de services infonuagiques entre 2007 à 2012. Particulièrement, nous avons utilisé Amazon comme référence qui affichait une disponibilité moyenne de 99.954 % pour les 5 années. De cette base d'entrainement, nous avons utilisé un tiers des données soit 250 enregistrements comme données de test.

4.2.1.3 Scénarios de test

Après échantillonnage, nous avons testé quatre scénarios, dont trois pour l'apprentissage des paramètres et 1 pour l'apprentissage de la structure afin de comparer lequel produira de meilleurs résultats.

Scénario 1 : nous avons utilisé un réseau bayésien naïf avec les données complètes et des variables discrètes sans paramètres manquants. Les valeurs discrètes de la variable DDS par exemple correspondent à 1=*bass*e lorsque la disponibilité dans le processus aléatoire est inférieure ou égale à 99.95 % et 2=*haute* en cas contraire

Scénario 2 : nous avons caché 20 % des paramètres du réseau naïf du scénario 1. Pour l'apprentissage des paramètres, nous avons utilisé l'algorithme EM (voir figure 15) qui évalue le maximum de vraisemblance avec une borne maximale de 15 itérations.

Scénario 3 : Nous avons construit un réseau mixte contenant de variables discrètes et continues pour lequel nous avons appris de nouveaux paramètres. Dans ce cas, nous avons posé comme hypothèse que la valeur de certaines variables sont considérées continues, avec les bornes [1=*basse*, 2=*haute*] et d'autres sont considérées discrètes.

Scénario 4 : nous avons fait l'apprentissage de la structure par l'algorithme K2 (Cooper and Herskovits 1992) qui serait le meilleur canevas pour les paramètres appris.

Les résultats de l'évaluation de ces quatre scénarios sont affichés dans une courbe ROC (*Receiver Operating Characteristic)* pour une meilleure appréciation. Une courbe ROC est une représentation graphique permettant de mesurer la performance d'un classificateur par rapport à un autre en fonction du risque associé (sensibilité/spécificité) (Cook 2007; Parker, et al. 2010).

4.2.2 Évaluation du module intelligence

L'ontologie probabiliste que nous avons construite apporte une solution efficace pour la prédiction de violation en tenant compte de l'incertitude sur certains paramètres. Les principales requêtes que nous avons testées fournissant des évidences permettent de répondre aux questions suivantes :

Requête 1 : en se basant sur les données historiques, quelle est la probabilité qu'une violation liée à la disponibilité survienne?

Évidence : Vérifier qu'une MV quelconque (p.ex., *MV1)* rapporte une disponibilité inférieure à 99.95 %.

Requête 2 : comment le temps de réponse influence la disponibilité d'une MV?

Évidence : vérifier que le temps de réponse des machines virtuelles *MV1* et *MV2* sont longs et le temps de réponse de *MV3* est court.

Requête 3 : une violation est-elle imminente par rapport aux résultats des deux requêtes précédentes?

Évidence : Disponibilité des MVs des résultats des requêtes 1 et 2.

Les résultats de test des évaluations de ces deux modules sont présentés et analysés dans le chapitre 5.

Chapitre 5 : Les résultats

Ce chapitre présente l'analyse des résultats obtenus après la mise en œuvre des évaluations décrites au chapitre précédent. Les résultats sont divisés en deux parties. La première partie concerne le module de confiance et la deuxième partie concerne le module de l'intelligence.

5.1 Résultats obtenus pour le module de confiance

Dans la première partie de notre expérimentation, nous avons implémenté trois types de réseaux (un réseau bayésien naïf discret, un réseau avec des paramètres manquants appris par l'algorithme EM et un réseau mixte) correspondants aux trois scénarios décrits dans le chapitre 4. Dans chacun de ces scénarios, nous avons calculé la probabilité que le niveau de confiance soit élevé (haut) avec hypothèse que les autres variables sont observées et connues. Cette probabilité est notée *P (NDC=haut | DDS, TDR, BPR, BPD, QDS).*

Les résultats obtenus sont affichés dans une courbe ROC basée sur une matrice de confusion qui donne en coordonnée le pourcentage de bonne prédiction et les fausses alertes en abscisse. Idéalement, une prédiction parfaitement efficace devrait avoir un taux de bonne prédiction proche de 100 % pour un taux de fausse alarme proche de 0 %.

5.1.1 Les paramètres appris

Dans notre cas, le réseau dont les paramètres sont appris par l'algorithme EM (en vert sur la figure 19) donne un meilleur taux de prédiction par rapport aux autres, en allant de 82 % de bonne prédiction pour 0 % de fausse alarme. Ensuite vient le réseau mixte (en bleu sur la figure 19) atteignant un taux de 98 % de bonne prédiction pour un taux de 40 de fausse alarme. Le réseau naïf (en rouge sur la figure 19) traine avec un taux de 90 % de bonne prédiction pour un taux de 38 % de fausse alarme.

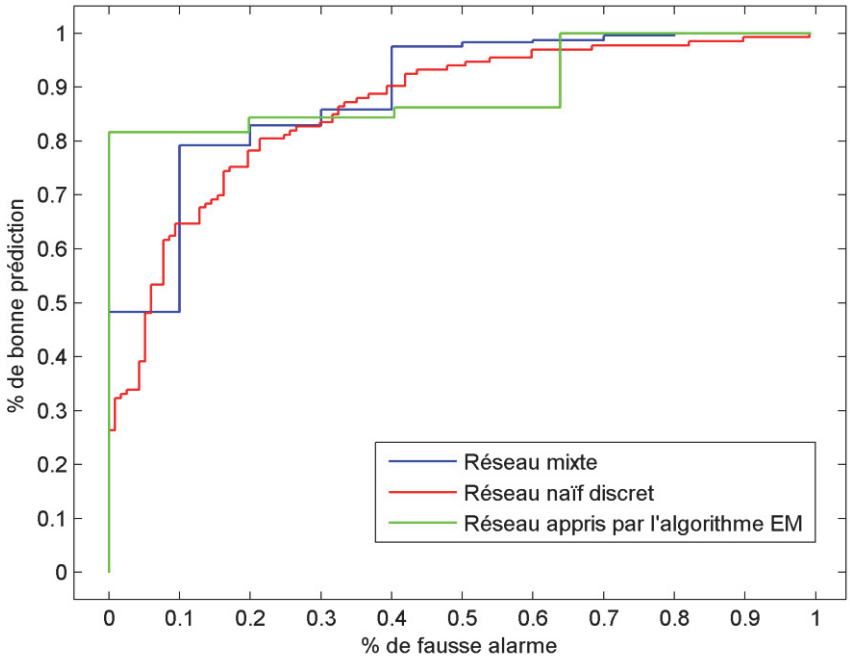


Figure 19 : Courbe ROC des paramètres appris suivant les scénarios

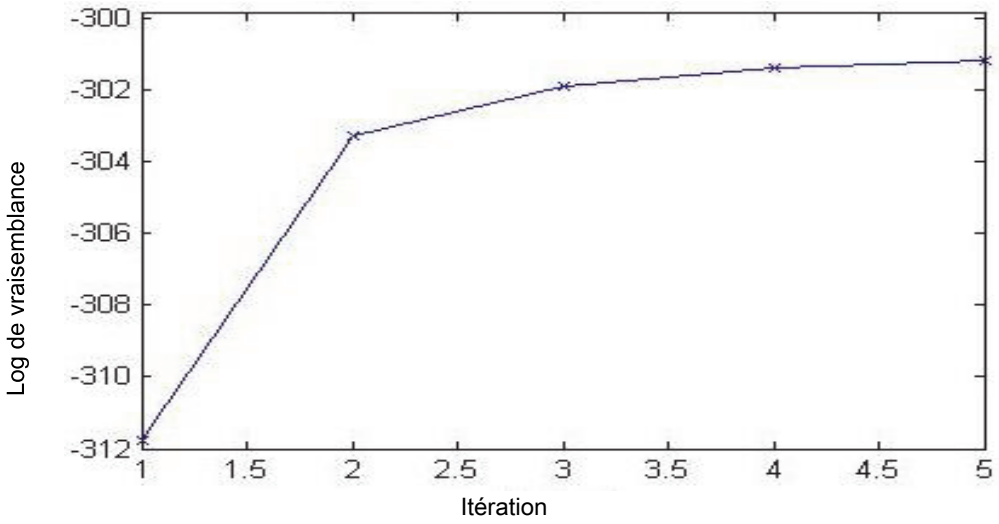


Figure 20 : Courbe log-vraisemblance pour les paramètres manquants par itération

La performance de l'algorithme EM ici (figure 20), avec une convergence à la cinquième itération (figure 20), peut s'expliquer par le fait que cet algorithme remplace les paramètres manquants par la valeur moyenne de l'espérance du réseau (voir le point a- de la section 3.2.1.1 du chapitre 3). Ainsi, dans le cas où la majorité des paramètres manquants se trouveraient en dessous de la valeur de l'espérance, l'algorithme EM peut offrir un avantage significatif par rapport aux autres méthodes d'apprentissage.

5.1.2 Les structures apprises

Dans la seconde partie de l'expérimentation de ce module, nous avons utilisé l'algorithme K2 pour apprendre la meilleure structure qui représenterait au mieux les paramètres appris dans la première partie. Les résultats obtenus sont sans équivoque, la structure apprise par l'algorithme K2 (figure 21) présente une forte hausse de taux de bonne prédiction pour un taux de fausse alarme; en particulier, nous avons 92 % de bonne prédiction pour moins de 10 % de fausse alarme et plus de 95 % de bonne prédiction pour de 20 % environ de fausse alarme.

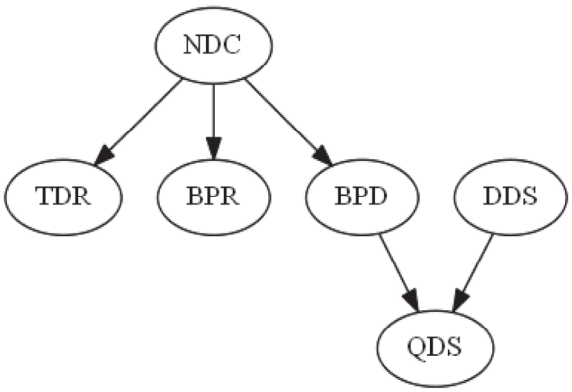


Figure 21 : Structure apprise par l'algorithme K2

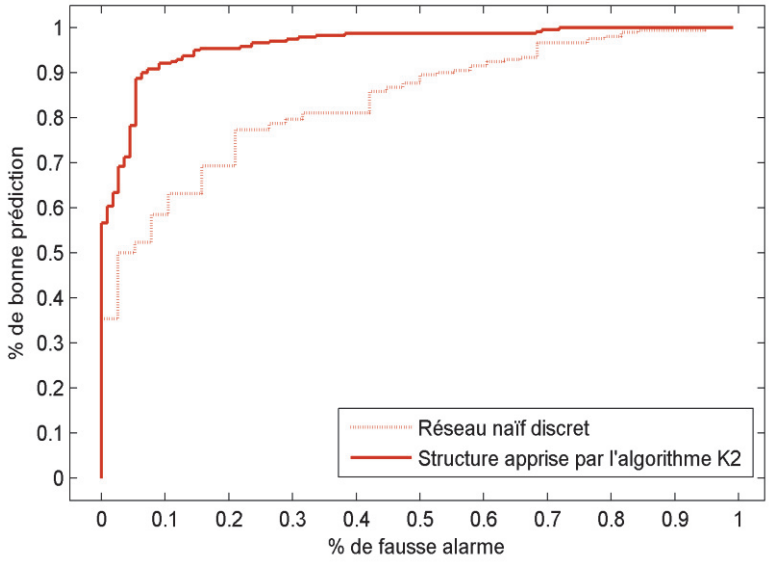


Figure 22 : Courbe ROC des résultats de la structure apprise par l'algorithme K2

L'efficacité de la structure apprise par l'algorithme K2 (Cooper and Herskovits 1992) peut s'expliquer par le fait que nous avons un réseau avec très peu de variables. Dans un cas d'un grand réseau avec une plus grande base de données, l'apprentissage par l'algorithme K2 (Cooper and Herskovits 1992) risque d'avoir une baisse exponentielle de son efficacité.

5.2 Résultats obtenus pour le module intelligence

Les résultats de l'évaluation de l'ontologie probabiliste sont présentés dans la figure 22. La partie gauche montre les requêtes et les évidences fournies à l'ontologie et à droite les résultats et les liens causaux qui font varier les probabilités des évènements liés aux évidences.

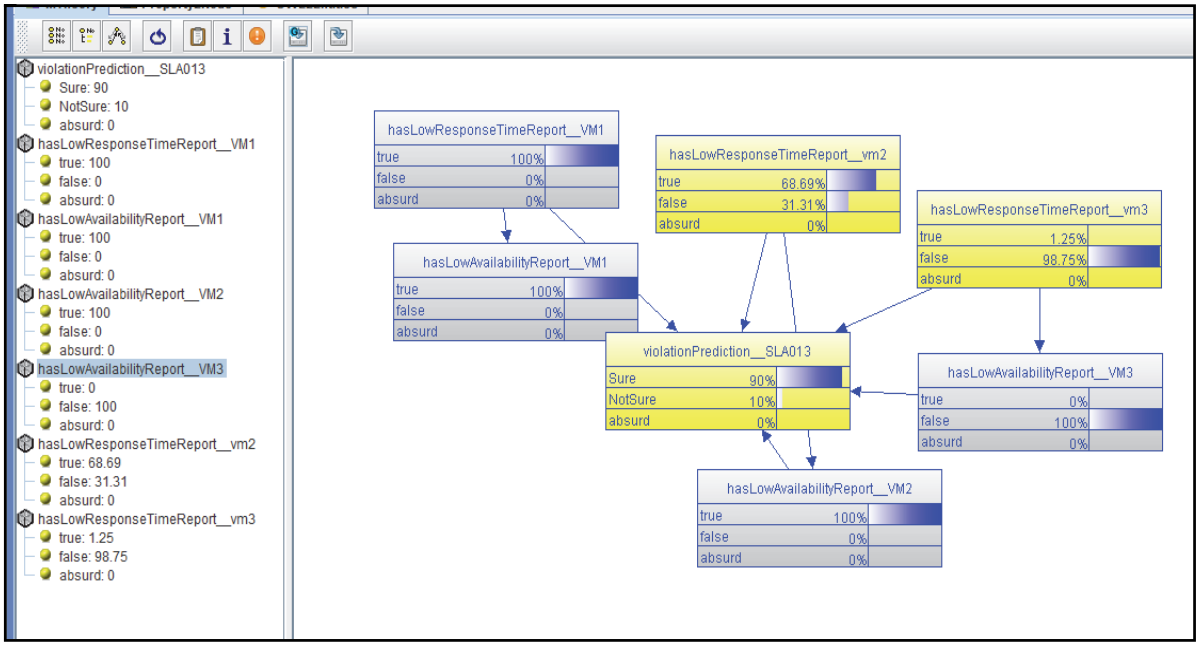


Figure 23 : Interface des résultats de l'ontologie probabiliste

Nous avons utilisé une référence de SLA (*SLA013*) pour lequel on a fourni des évidences sur la disponibilité des MVs (*VM1, VM2, VM3*) *hasLowAvailabilityReport_* et le temps de réponse *hasLowResponseTimeReport* afin de voir comment ceux-là pourraient affecter la violation. Les résultats montrent un taux de prédiction sûre à 90 % qui auraient pu être 100 % au cas où la VM3 n'aurait pas affiché une disponibilité à 100 %.

5.3 Conclusion

En général, les résultats obtenus au cours de nos simulations sont assez bons. Il s'agit d'une efficacité qui est due en partie à la taille du réseau et la petite quantité d'enregistrements dans la base de données de test. Par contre, l'application de notre démarche dans un environnement réel de services infonuagiques aura requis, dans le cas du module de confiance en particulier, la disponibilité des fichiers de sessions des fournisseurs concernés à une entité tierce ou un courtier par exemple. Vu la sensibilité des fichiers de sessions, cela pourrait poser des problèmes de sécurité et de confidentialité

d’une part. D’autre part, cela aura requis des algorithmes plus robustes pour les moteurs d’inférence pour les données de grandes tailles.

Une solution à ce problème serait de limiter la taille des données historiques pour ne pas trainer toute une base de fichiers de sessions à chaque mise à jour de la confiance en introduisant un facteur d’oubli proposé par Josang et Ismail (Jsang and Ismail 2002) consistant à se limiter à un seuil N des n derniers interactions.

Nous n’avons pas pu faire des comparaisons à cause notre approche pour le module de confiance est la première de ce genre dans l’environnement infonuagique. Nous n’avons pas pu faire des comparaisons pour le module d’intelligence non plus, car les autres travaux utilisant une ontologie déterministe n’ont pas pu quantifier les résultats obtenus,

Chapitre 6 : Conclusion générale et perspective

L'infonuage est l'un des nouveau-nés des services informatiques disponibles à la demande dans Internet. Il a connu un grand essor caractérisé par la dématérialisation des ressources et un modèle de paiement régi par un contrat, le SLA.

L'application du SLA dans l'environnement infonuagique est un problème complexe vu la nature dynamique de l'infonuage. L'incertitude qui règne autour des infrastructures du système infonuage et l'incertitude sur le comportement du client en terme de requête rendent de plus en plus difficile le respect des clauses du SLA par le fournisseur.

Il devient donc impératif pour un client, de choisir un fournisseur de confiance, ayant une bonne réputation, mais aussi qui apriori, peut garantir le respect de ses engagements dans un futur proche avant de négocier le contrat. En même temps, le fournisseur doit être en mesure d'anticiper les violations, surtout en ce qui a trait à la disponibilité du service, afin de garantir une disponibilité élevée et améliorer son niveau de confiance. Dans cet objectif, plusieurs travaux ont proposé des solutions pour calculer le niveau de confiance d'un fournisseur en tenant compte uniquement des données historiques sans une perspective de confiance vers l'avenir. Pour ce qui a trait aux violations, certains travaux ont appliqué une ontologie probabiliste, qui à partir des règles détermine les cas de violations sans la capacité de les prédire encore moins, de tenir compte de l'incertitude qu'on peut avoir autour des données.

Dans ce mémoire, nous avons proposé un cadre d'applications qui, d'une part, fait une prédiction sur le niveau de confiance d'un fournisseur en se basant sur les données historiques de ce dernier. Même en cas de données manquantes notre module de confiance basé sur un réseau bayésien arrive à faire une prédiction efficace de la confiance soit 82 de bonne prédiction pour 0 de fausse alarme. En faisant un apprentissage de la structure par l'algorithme K2 (Cooper and Herskovits 1992) (Robinson 1977), le taux de bonne prédiction atteint 92 % de bonne prédiction pour moins de 10 % de fausse alarme. Il est conseillé dans le cadre des prédictions de choisir le taux de bonne prédiction assez élevé qui correspond à un taux de fausse alarme le plus proche de 0 possible. Dans notre cas,

l'algorithme EM appliquée aux données manquantes semble, apriori, produire de meilleurs résultats. Par contre, en tenant compte des données complètes, sans paramètres manquants, utilisés dans par l'algorithme K2 (Cooper and Herskovits 1992) (Robinson 1977) nous avons donné plus de crédibilité aux résultats produits par ce dernier.

Dans un second temps, nous avons utilisé une ontologie probabiliste, PR-OWL, en plus d'être compatible avec OWL, cette ontologie basée sur les réseaux bayésiens à entités multiples est capable de prédire tout en tenant compte des incertitudes autour des données. Les résultats montrent la fiabilité de notre approche avec une probabilité de 90 % de prédiction en se basant sur les évidences.

Il est prouvé que les réseaux bayésiens sont des méthodes fiables pour faire des prédictions en se basant sur des faits réels même en présence d'incertitudes autour de ces faits. Nous avons conclu que les réseaux bayésiens peuvent être appliqués dans l'environnement infonuagique pour prédire le niveau de confiance d'un fournisseur et les violations.

Notre travail se limite, d'une part, par les outils utilisés. Premièrement, les algorithmes utilisés dans BNT par exemple affecteraient les résultats à la baisse en présence d'une grande quantité de données. Deuxièmement, il existe très peu de documentation sur l'utilisation de Unbbayes, en plus, il n'y a qu'une petite équipe travaillant à son évolution. D'autre part, le nombre de variables utilisées dans le réseau bayésien ne permettent pas de prendre en considération tous les aspects et détails des fichiers de session qui auraient pu aider à faire des prédictions plus approfondies. Les données utilisées dans les tests sont obtenues par simulation, donc elles risquent ne pas refléter parfaitement la situation réelle de 'environnement infonuagique.

À l'avenir, nous comptons développer et tester notre cadre d'applications dans un environnement réel d'infonuage malgré les risques de sécurité qui peuvent être liés à notre approche en partageant les fichiers de sessions avec d'autres entités d'une part. D'autre part, nous espérons utiliser des outils plus robustes capables de produire des résultats aussi efficaces avec un grand nombre de variables qu'avec un grand nombre d'enregistrements.

Références bibliographiques

Bauer, Eric, and Randee Adams

2012 Cloud Computing. *In* Reliability and Availability of Cloud Computing. Pp. 1-15 : John Wiley & Sons, Inc.

Berners-Lee, Tim, and Mark Fischetti

2000 Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor: HarperInformation.

Buyya, R., S. K. Garg, and R. N. Calheiros

2011 SLA-oriented resource provisioning for cloud computing: Challenges, architecture, and solutions. 2011 International Conference on Cloud and Service Computing (CSC 2011), 12-14 Dec. 2011, Piscataway, NJ, USA, 2011, pp. 1-10. IEEE.

Carvalho, RommelN, KathrynB Laskey, and PauloC G. Costa

2013 PR-OWL 2.0 – Bridging the Gap to OWL Semantics. *In* Uncertainty Reasoning for the Semantic Web II. F. Bobillo, P.G. Costa, C. d'Amato, N. Fanizzi, K. Laskey, K. Laskey, T. Lukasiewicz, M. Nickles, and M. Pool, eds. Pp. 1-18. Lecture Notes in Computer Science : Springer Berlin Heidelberg.

Chakraborty, S., and K. Roy

2012 An SLA-based Framework for Estimating Trustworthiness of a Cloud. Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, 2012, pp. 937-942.

Chen, Zhou, Chia Liang-Tien, and Lee Bu-Sung

2004 DAML-QoS ontology for Web services. Web Services, 2004. Proceedings. IEEE International Conference on, 2004, pp. 472-479.

Chieu, T. C., et al.

2009 Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment. e -Business Engineering, 2009. ICEBE '09. IEEE International Conference on, 2009, pp. 281-286.

Cook, Nancy R

2007 Use and misuse of the receiver operating characteristic curve in risk prediction. Circulation 115(7) : 928-935.

Cooper, Gregory F, and Edward Herskovits

1992 A Bayesian method for the induction of probabilistic networks from data. Machine learning 9(4) : 309-347.

Costa, Paulo CG

2005 Bayesian semantics for the Semantic Web: George Mason University.

Costa, PauloCesarG, KathrynB Laskey, and KennethJ Laskey

2008 PR-OWL : A Bayesian Ontology Language for the Semantic Web. *In* Uncertainty Reasoning for the Semantic Web I. P. Costa, C. d'Amato, N. Fanizzi, K. Laskey, K. Laskey, T. Lukasiewicz, M. Nickles, and M. Pool, eds. Pp. 88-107. Lecture Notes in Computer Science : Springer Berlin Heidelberg.

d'Ambrosio, Bruce

1991 Local expression languages for probabilistic dependence: a preliminary report. Proceedings of the Seventh conference on Uncertainty in Artificial Intelligence, 1991, pp. 95-102. Morgan Kaufmann Publishers Inc.

Dempster, Arthur P, Nan M Laird, and Donald B Rubin

1977a Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (Methodological) : 1-38.

1977 b Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal statistical Society 39(1) : 1-38.

Deutsch, Morton

1962 Cooperation and trust: Some theoretical notes.

Ding, Zhongli, Yun Peng, and Rong Pan

2006 BayesOWL: Uncertainty Modeling in Semantic Web Ontologies. *In* Soft Computing in Ontologies and Semantic Web. Z. Ma, ed. Pp. 3-29. Studies in Fuzziness and Soft Computing: Springer Berlin Heidelberg.

Dobson, G., R. Lock, and I. Sommerville

2005 QoSOnt: a QoS ontology for service-centric systems. Software Engineering and Advanced Applications, 2005. 31 st EUROMICRO Conference on, 2005, pp. 80-87.

Emeakaroha, V. C., et al.

2010 Low level Metrics to High level SLAs - LoM2HiS framework: Bridging the gap between monitored metrics and SLA parameters in cloud environments. High Performance Computing and Simulation (HPCS), 2010 International Conference on, 2010, pp. 48-54.

Emeakaroha, V. C., et al.

2012 Towards autonomic detection of SLA violations in Cloud infrastructures. Future Generation Computer Systems-the International Journal of Grid Computing and Escience 28(7) : 1017-1029.

Fakhfakh, K., et al.

2008 A Comprehensive Ontology-Based Approach for SLA Obligations Monitoring. Advanced Engineering Computing and Applications in Sciences, 2008. ADVCOMP '08. The Second International Conference on, 2008, pp. 217-222.

Firdhous, M., O. Ghazali, and S. Hassan

2011 A trust computing mechanism for cloud computing with multilevel thresholding. Industrial and Information Systems (ICIIS), 2011 6th IEEE International Conference on, 2011, pp. 457-461.

Gagnaire, Maurice, et al.

2012 Downtime statistics of current cloud solutions. International Working Group on Cloud Computing Resiliency, Tech. Rep., June:176-189.

Gambetta, D.

2000 Trust Making and Breaking Cooperative Relations. Departement of Sociologie, University of Oxford (Can we Trust Trust) : 213-237.

Hagen, S., M. Seibold, and A. Kemper

2012 Efficient verification of IT change operations or: How we could have prevented Amazon's cloud outage. Network Operations and Management Symposium (NOMS), 2012 IEEE, 2012, pp. 368-376.

Heckerman, David

1997 Bayesian Networks for Data Mining. Data Mining and Knowledge Discovery 1(1) : 79-119.

Hogan, Michael, et al.

2011 Nist cloud computing standards roadmap. NIST Special Publication 35.

Hussain, F. K., O. K. Hussain, and E. Chang

2007 An overview of the interpretations of trust and reputation. Emerging Technologies and Factory Automation, 2007. ETFA. IEEE Conference on, 2007, pp. 826-830.

Jøsang, Audun, Roslan Ismail, and Colin Boyd

2007 A survey of trust and reputation systems for online service provision. Decision Support Systems 43(2) : 618-644.

Jøsang, Audun, and Simon Pope

2005 Semantic constraints for trust transitivity. Proceedings of the 2nd Asia-Pacific conference on Conceptual modelling-Volume 43, 2005, pp. 59-68. Australian Computer Society, Inc.

Jsang, Audun, and Roslan Ismail

2002 The beta reputation system. Proceedings of the 15th bled electronic commerce conference : 41-55.

Laskey, Kathryn Blackmond, et al.

2011 PR-OWL 2 case study : A maritime domain probabilistic ontology. Proceedings of the 6th International Conference on Semantic Technologies for Intelligence, Defense, and Security, Fairfax, VA, 2011, pp. 76-83.

Lauritzen, S. L., and D. J. Spiegelhalter

1988 Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. Journal of the Royal Statistical Society. Series B (Methodological) 50(2) : 157-224.

Lauritzen, Steffen L

1992 Propagation of probabilities, means, and variances in mixed graphical association models. Journal of the American Statistical Association 87(420) : 1098-1108.

Liu, Hui, Fenglin Bu, and Hongming Cai

2012 SLA-based service composition model with semantic support. 2012 7th IEEE Asia-Pacific Services Computing Conference, APSCC 2012, December 6, 2012 - December 8, 2012, Guilin, China, 2012, pp. 374-379. IEEE Computer Society.

Luhmann, Niklas

2000 Familiarity, confidence, trust : Problems and alternatives. Trust : Making and breaking cooperative relations 6:94-107.

Macias, M., and J. Guitart

2012a Client Classification Policies for SLA Negotiation and Allocation in Shared Cloud Datacenters. Economics of Grids, Clouds, Systems, and Services. 8 th International Workshop, GECON 2011, 5 Dec. 2011, Berlin, Germany, 2012a, pp. 90-104. Springer-Verlag.

Macias, Mario, and Jordi Guitart

2012 b Client classification policies for SLA enforcement in shared cloud datacenters. 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2012, May 13, 2012 - May 16, 2012, Ottawa, ON, Canada, 2012 b, pp. 156-163. IEEE Computer Society.

Marsh, Stephen Paul

1994 Formalising trust as a computational concept, School of Natural Sciences, University of Stirling.

MathWorks

2013 MATLAB R2013a and BNT Toolbox Natick, Massachusetts, United States.: The MathWorks, Inc.

Maurer, Michael, et al.

2012 Cost-benefit analysis of an SLA mapping approach for defining standardized Cloud computing goods, P.O. Box 211, Amsterdam, 1000 AE, Netherlands, 2012. Vol. 28, pp. 39-47. Elsevier.

Michael Maurer , Vincent Emeakaroha

2012 Governance of Cloud Computing Infrastructures using Knowledge Management" and on "Managing Cloud Service Provisioning and SLA Enforcement via Holistic Monitoring Techniques.

Motik, Boris, et al.

2009 Owl 2 web ontology language : Profiles. W3C recommendation 27:61.

Naïm, Patrick, et al.

2011 Réseaux bayésiens : Editions Eyrolles.

Newsroom, Gatner

2013 Iaas continues as fastest-growing market segment. *In* Gartner Says Worldwide Public Cloud Services Market to Total $131

Billion. Pp. 7. STAMFORD, Conn.: Gartner Inc.

Parker, D. R., et al.

2010 Development of a Bayesian Framework for Determining Uncertainty in Receiver Operating Characteristic Curve Estimates. Knowledge and Data Engineering, IEEE Transactions on 22(1) : 31-45.

Qiang, Guo, et al.

2011 Modeling and evaluation of trust in cloud computing environments. Advanced Computer Control (ICACC), 2011 3rd International Conference on, 2011, pp. 112-116.

Research, Stanford Center for Biomedical Informatics

Protegé ontology editor and knowledge acquisition system http://protege.stanford.edu.

Robinson, Robert W

1977 Counting unlabeled acyclic digraphs. *In* Combinatorial mathematics V. Pp. 28-43 : Springer.

Shachter, Ross D, and C Robert Kenley

1989 Gaussian influence diagrams. Management science 35(5) : 527-550.

Shadbolt, N., W. Hall, and T. Berners-Lee

2006 The Semantic Web Revisited. Intelligent Systems, IEEE 21(3) : 96-101.

Sturm, Rick, Wayne Morris, and Mary Jander

2000 Foundations of service level management. Volume 13 : Sams Indianapolis, IN.

Sun, Yih Leong, et al.

2010 SLA-aware Resource Management. *In* Grids and Service-Oriented Architectures for Service Level Agreements. P. Wieder, R. Yahyapour, and W. Ziegler, eds. Pp. 35-44 : Springer US.

Vu, QuangHieu, Mihai Lupu, and BengChin Ooi

2010 Trust and Reputation. *In* Peer-to-Peer Computing. Pp. 183-214 : Springer Berlin Heidelberg.

Vuong Xuan, Tran

2008 WS-QoSOnto: A QoS Ontology for Web Services. Service-Oriented System Engineering, 2008. SOSE '08. IEEE International Symposium on, 2008, pp. 233-238.

W3C, World Wide Web Consortium

2014 RDF 1.1 Primer. (W3C Working Group Note 25 February 2014).

Wieder, et al.

2011 Service level agreements for cloud computing.

Wu, Linlin, Saurabh Kumar Garg, and Rajkumar Buyya

2011 SLA-based resource allocation for software as a service provider (SaaS) in cloud computing environments. 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2011, May 23, 2011 - May 26, 2011, Newport Beach, CA, United states, 2011, pp. 195-204. IEEE Computer Society.

Yong Beom, Ma, Jang Sung Ho, and Lee Jong Sik

2011 Ontology-Based Resource Management for Cloud Computing. Intelligent Information and Database Systems. Third International Conference, ACIIDS 2011, 20-22 April 2011, Berlin, Germany, 2011. Vol. pt. II, pp. 343-52. Springer Verlag.

Zhang, Qi, Lu Cheng, and Raouf Boutaba

2010 Cloud computing: state-of-the-art and research challenges. Journal of internet services and applications 1(1) : 7-18.