

# Thèse Professionnelle

## La GRC sur Internet : Application du Web Sémantique à un Conseiller Interactif



Responsable (ICAME E76) : Stéphane Rouilly  
Responsable (ICAME E77) : Pascale Capron  
Responsable (Devoteam) : Jean-Guillaume Buvry

Responsable MS : Isabelle Demeure  
Tuteur : Georges Hebrail



## Remerciements

Je tiens chaleureusement à remercier :

- Stéphane Rouilly et Pascale Capron, responsables de l'action « Conseiller Interactif » pour leur accueil chaleureux, leur professionnalisme et leur disponibilité
- Pierre Chaussecourte, ingénieur chercheur dans le groupe E76, pour sa confiance
- Isabelle Demeure, responsable du MS CASI, qui m'a permis de réaliser mon rêve d'intégrer Télécom Paris
- Georges Hebrail, professeur à Télécom Paris et tuteur de ma thèse professionnelle
  
- Isabelle Vautier, pour sa collaboration
- Jean-Guillaume Buvry, ingénieur d'affaire chez Devoteam, pour sa confiance
- Laurent Olivry, ingénieur de recherche dans le département SINETICS, pour sa collaboration
- Franck Martin, consultant pour SINETICS, pour sa collaboration
- Sylvaine Nugier, du SOAD, pour sa gentillesse
  
- Toute l'équipe du groupe E76
- SL et Vé, mes parents, pour leur support
- Valérie, Fei-Ching et Joan, pour leur soutien

# Plan

I.	Introduction .....	5
II.	Contexte.....	6
A.	Le Groupe EDF .....	6
1.	Stratégie et structure .....	6
2.	La Division Recherche et Développement .....	6
3.	Solutions et technologies innovantes pour l'offre commerciale.....	8
B.	La gestion de la relation client .....	8
1.	Les canaux traditionnels.....	10
2.	Le canal Internet.....	11
3.	Les systèmes d'informations.....	12
4.	Appui GRC .....	14
C.	Stratégie et organisation.....	14
III.	Objectifs du Conseiller Interactif.....	16
A.	Description de l'action .....	16
B.	L'existant : le site Web « Particuliers » et les services interactifs.....	17
1.	Le site Web « Particuliers » .....	17
2.	Les services interactifs .....	18
3.	Architecture globale à mettre en place.....	19
C.	Synthèse de ce qui a été fait.....	19
IV.	Les technologies.....	20
A.	Les solutions clé en main.....	20
1.	Les outils d'e-CRM clés en main.....	21
2.	Fonctionnalités à retenir pour notre plateforme .....	24
3.	Les BRMS : moteurs de règles .....	25
B.	Le Web Sémantique.....	30
1.	Le XML .....	30
2.	RDF.....	31
3.	OWL .....	32
4.	Couches du Web Sémantique.....	33
5.	Avantages « théoriques » du WS .....	34
6.	Les outils.....	34
7.	Les raisonneurs.....	43
8.	Les protocoles de communications.....	46
V.	Le Conseiller Interactif.....	49
A.	Cahier des charges .....	49
B.	La plateforme du Conseiller Interactif.....	49
C.	Architecture métier.....	50
D.	Choix des technologies .....	51
1.	Architecture du Conseiller .....	51
2.	Serveur d'application .....	51
3.	La base de données relationnelle .....	52

E.	Modélisation des connaissances .....	52
1.	Le modèle « Internaute » .....	52
2.	Une ontologie sur le contenu du site Internet .....	53
3.	Les règles marketing .....	53
F.	Le veilleur.....	55
1.	Identification.....	55
2.	Collecte d'informations .....	55
3.	Le composant de veille.....	56
G.	Le tableau de bord.....	57
H.	Le conseiller.....	58
I.	Le raisonneur.....	59
J.	Les outils .....	60
1.	Visualisation des données et des déductions.....	60
2.	L'orgue de barbarie .....	61
3.	La console d'administration.....	62
K.	L'architecture de la plateforme .....	63
1.	Infrastructure technique.....	63
2.	Technologies .....	64
3.	Architecture des données.....	64
4.	Architecture finale.....	65
VI.	Conclusion.....	66
A.	Bilan de la plateforme .....	66
1.	Mesure de performances.....	66
2.	Etude de la scalabilité Web .....	68
B.	Apports du Web Sémantique.....	70
1.	Résolution du problème avec une ontologie.....	70
2.	Bénéfices de l'utilisation d'une ontologie .....	73
3.	Conclusion sur le Web Sémantique.....	75
C.	Perspectives .....	76
D.	Epilogue .....	76
VII.	Bibliographie .....	77

## Illustrations

Figure 1 - Dynamique de la GRC .....	9
Figure 2 - Architecture de l'Agence En Ligne.....	12
Figure 3 - Organisation de l'action .....	14
Figure 4 - Livrables .....	15
Figure 5 - Architecture générale du site Internet.....	17
Figure 6 - Architecture à mettre en place .....	19
Figure 7 - Architecture de Vignette Dialog .....	22
Figure 8 - E-deal Automation.....	23
Figure 9 - Un moteur de règles : comment ça marche ?.....	25
Figure 10 - Quelques moteurs de règles.....	26
Figure 11 - Jrules + J2EE.....	27
Figure 12 - Drools.....	28
Figure 13 - Protégé.....	39
Figure 14 - OilEd .....	41
Figure 15 - Semantic Works .....	42
Figure 16 - La séparation des acteurs et des référentiels .....	50
Figure 17 - Ontologie "Page" .....	53
Figure 18 - Le veilleur .....	56
Figure 19 - Le tableau de bord .....	57
Figure 20 - Le conseiller .....	58
Figure 21 - Le conseiller sur une page Web .....	59
Figure 22 - Le raisonneur .....	59
Figure 23 - Outil de visualisation du raisonnement (GRaison) .....	60
Figure 24 - GRaison : détails d'une règle .....	61
Figure 25 - Outil de datamining (GEnrichis).....	61
Figure 26 - La console d'administration.....	62
Figure 27 - Infrastructure technique du Conseiller Interactif.....	63
Figure 28 - ODP technologie .....	64
Figure 29 - Référentiels.....	64
Figure 30 - Vue générale de la plateforme (ODP ingénierie) .....	65

# I. Introduction

Pour la thèse professionnelle de mon Mastère Spécialisé en Conception et Architecture des Systèmes Informatique de l'Ecole Nationale Supérieure des Télécommunications, j'ai eu l'opportunité d'effectuer une mission longue au département ICAME de la Division Recherche et Développement d'EDF. Ayant effectué mon stage de Master en Ingénierie Informatique de l'Université Paris XI en 2004 dans le groupe Statistiques et Outils d'Aide à la Décision de ce même département dans d'excellentes conditions ; je n'ai pas hésité un instant à réaliser cette mission en tant que Consultant du Groupe Devoteam.

La mission consistait à appuyer le groupe chargé de la Gestion de la Relation Client pour l'élaboration d'un conseiller interactif pour le site Web des particuliers EDF. Il m'a alors été demandé de concevoir et développer une plateforme modulaire portant le nom du « Conseiller Interactif ».

La question que nous nous posons avant l'élaboration d'un prototype est la suivante : « Le Web Sémantique peut-il être appliqué pour de la GRC ? ».

Nous allons tâcher de répondre à cette question tout au long de cette thèse professionnelle. Dans un premier lieu, en étudiant le contexte de la GRC chez EDF puis en effectuant un inventaire des besoins qui nous mènent à élaborer un « Conseiller Interactif ». Nous allons par la suite faire un panorama des solutions et des technologies applicables. Technologies que nous allons utiliser pour l'élaboration d'un prototype du « Conseiller Interactif ». Nous analyserons ce prototype et finalement, nous ferons un bilan technique de cette plateforme et un point sur les apports du Web Sémantique.

## II. Contexte

Je vais, ici, vous décrire la structure EDF, groupe où j'ai réalisé ma thèse professionnelle. Cette thèse a été réalisée dans la Division Recherche et Développement et s'intègre dans le cursus de Mastère Spécialisé en Conception et Architecture des Systèmes Informatiques de l'Ecole Nationale Supérieure des Télécommunications de Paris.

### ***A. Le Groupe EDF***

#### **1. Stratégie et structure**

Le Groupe est un des leaders européens de l'énergie. Sa croissance s'oriente vers de nouveaux marchés et de nouvelles activités. Le principal marché d'EDF reste le territoire français mais le Groupe étend sa stratégie dans un contexte de concurrence internationale. L'ouverture du marché de l'énergie à la concurrence est, pour EDF, un défi et l'opportunité de devenir un énergéticien de référence en Europe. En France, le marché énergétique a été ouvert dès 1999, avec une première tranche de clients éligibles<sup>2</sup> (consommation électrique minimale de 100 GWh par an) élargie en janvier 2000 à 30% du marché (consommation minimale de 16 GWh par an). Depuis le 1er juillet 2004, le marché de l'électricité est ouvert à tous les clients professionnels ; et le sera le 1er juillet 2007 pour les particuliers. La France se prépare ainsi à l'ouverture à 100% de son marché de l'électricité à la concurrence. L'évolution d'EDF a été initiée depuis plusieurs années par la mise en place d'une nouvelle organisation qui devrait favoriser également le développement de nouveaux métiers et de nouveaux services permettant ainsi au Groupe de demeurer l'un des leaders du marché européen.

Le groupe EDF répond à l'engagement commun de fourniture d'énergie au service de l'intérêt général. Les valeurs éthiques du groupe guident les actions de chacun : le respect de la personne et de l'environnement, la performance, la solidarité et l'intégrité. Les valeurs du groupe impliquent de la part des sociétés d'EDF l'application individuelle et collective de principes d'action afin de répondre au mieux aux attentes des clients, des actionnaires, des fournisseurs, des collectivités...

#### **2. La Division Recherche et Développement**

La Division EDF, Recherche et Développement (EDF R&D), dépend de la Direction Générale des Opérations. Elle est répartie sur trois sites principaux situés en Île-de-France : Chatou (78), Les Renardières (77) et Clamart (92).

Ma thèse professionnelle s'est effectuée sur le site de Clamart, qui regroupe la moitié des forces de la recherche d'EDF. Il réunit surtout des activités à caractère non expérimental et des moyens de calculs relatifs à l'informatique scientifique. Le pôle Recherche et Développement de EDF comporte pas moins de cinq programmes globaux :

- Réseau et environnement,
- Management d'énergies,
- Développement commercial,
- Production,
- Technologies de l'information.



La branche R&D travaille également sur des domaines parallèles tels que les technologies de l'information et de l'environnement. Elle est constituée de 16 départements ; et j'ai réalisé ma mission dans le département Innovation Commerciale, Analyse des Marchés et de leur Environnement (ICAME). Il se situe au sein du programme « Développement commercial ».

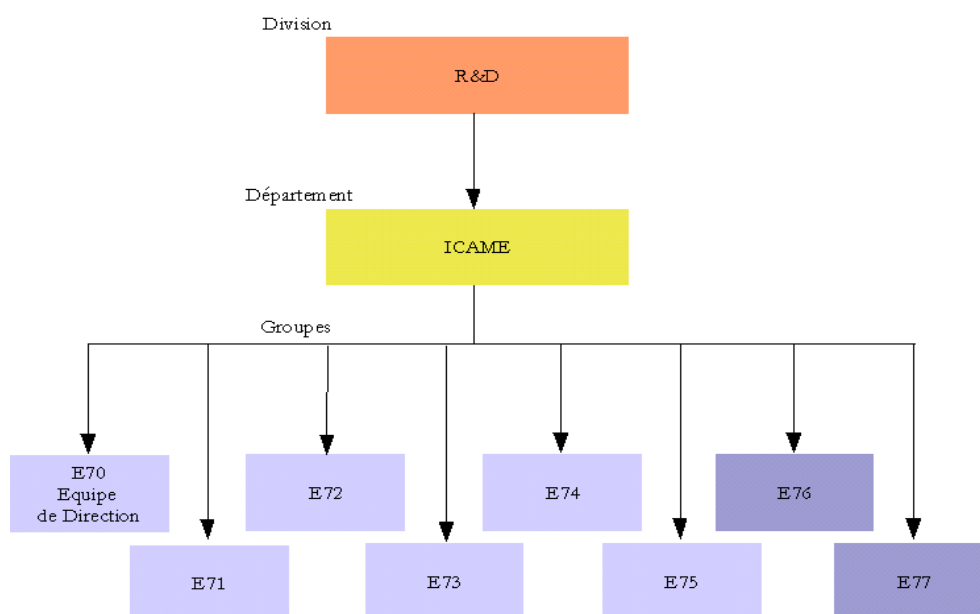
Les enjeux de ces programmes visent des clients dont le rapport à l'entreprise est amené à évoluer dans un avenir très proche. En effet, tous les clients particuliers d'EDF, en France, pourront adopter un tarif non régulé d'ici à juillet 2007, date de l'ouverture de ce dernier secteur encore régulé. Cela nécessite donc une révision en profondeur de l'offre de fourniture d'énergie. Ainsi, la R&D ne joue pas seulement un rôle d'innovation ou de recherche appliquée. Beaucoup d'agents y travaillent afin de mieux cerner les clients d'EDF (près de 22 millions de particuliers et de 2,2 millions de professionnels en France), de les fidéliser et de comprendre leurs demandes en matière de services. Cela se traduit par un observatoire des consommations et de satisfaction des clients, un enrichissement des bases de données, etc....

En outre, le programme « Développement commercial » imagine les offres d'énergie de demain ainsi que les services en matière de confort dans le logement, en matière de sécurité et d'efficacité énergétique, au même titre que les services à distance qui s'appuieront largement sur les nouvelles technologies comme Internet ou le téléphonie mobile.

Le département ICAME, créé début 2002, propose et élabore essentiellement de nouvelles offres de fourniture d'énergie et de services, intégrant la façon de les mettre en œuvre. Il joue ainsi un rôle d'appui à la commercialisation en lui proposant une compréhension élargie du lien entre EDF et ses marchés.

ICAME est divisé en neuf groupes (de E71 à E79), qui gèrent différents projets. Certains de ces groupes sont à dominante « métier » dédiés à un marché particulier (par exemple le marché résidentiel), d'autres à dominante « compétences » chargés de développer de nouvelles méthodes.

Voici les groupes E76<sup>1</sup> et E77<sup>2</sup> au sein de l'organigramme EDF R&D :



<sup>1</sup> Solutions et technologies innovantes pour l'Offre Commerciale

<sup>2</sup> Innovation et Processus de Conception

### 3. Solutions et technologies innovantes pour l'offre commerciale

J'ai effectué ma mission dans le groupe E76 - Solutions et technologies innovantes pour l'Offre Commerciale. Ce groupe s'occupe de la conception d'offres de services qui demande une connaissance et une appropriation des technologies qui serviront de support pour porter le service jusqu'au consommateur.

Cette vision, davantage technique, est intégrée dans ce groupe. Mensuellement ont lieu des réunions de groupes pour faire un état d'avancement de ces projets, mais surtout pour régler tous les problèmes administratifs et logistiques inhérents au groupe. Ces réunions impliquent la participation de tous les agents du groupe en question mais aussi des prestataires, cas dans lequel je me trouve lors de l'accomplissement de cette mission.

Toutes les idées sont puisées dans l'observation de la compréhension des clients sous leurs différentes facettes, qu'ils soient consommateurs ou citoyens. Mon sujet de thèse professionnelle vient dans cette idée de comprendre et connaître le client. De nombreux sujets visant à augmenter l'interactivité ou la personnalisation du site pour augmenter et fidéliser les visiteurs se développent chaque année. Ainsi, début 2005, arrive le projet « Conseiller Interactif ». Ce projet vient appuyer la Gestion de la Relation Client chez EDF.

#### ***B. La gestion de la relation client***

La Gestion de la Relation Client (GRC) trouve son fondement sur la connaissance du client ; il est souvent nécessaire de mettre en œuvre une base de données marketing ou même un entrepôt de données afin de stocker les informations sur ce dernier. Un des principaux facteurs de réussite de l'entreprise est sa capacité à exploiter les données de son système d'informations « client ».

L'analyse de ces bases de données permet l'établissement d'une stratégie pour la GRC : celle-ci permet alors d'obtenir de nouvelles informations sur le client qui enrichiront notre base de données et permettront de répondre à différentes questions.

Voici un tableau<sup>3</sup> résumant les objectifs de l'entreprise et les questions qu'elle doit se poser :

Objectif	Question
Acquérir des clients	Où trouver le nouveau client ?
Equiper les clients	Comment vendre plus au même client ?
Satisfaire ses clients	Comment intensifier l'utilisation des produits ou services ?
Fidéliser ses clients	Comment conserver la relation ?

<sup>3</sup> La Gestion de la relation client – Ed Eyrolles 2005 – R. Lefébure, G. Venturi – Page 113

Connaître le client implique quatre concepts de base :

1. La segmentation
2. Les techniques de scoring
3. Les outils de Datamining
4. La valeur client

Le S.I. sera alors le cœur de la stratégie marketing, les différents outils (scoring ou datamining) viendront puiser les informations dans ce système et nous permettront de segmenter les clients et de déterminer leurs « valeurs ».

La construction de Systèmes d'Informations autour du client transite par divers moyens de communication. Ces derniers seront les sources de données dont il faudra évidemment filtrer afin de ne retenir que les informations nécessaires.

La GRC chez EDF est « l'optimisation et la valorisation de chaque contact client grâce à une stratégie du « sur-mesure » appuyée sur la technologie pour l'industrialiser ».

Comme le montre le schéma ci-dessous, la GRC chez EDF repose sur 5 étapes qui gravitent autour du client. L'objectif est de créer une dynamique qui permettrait de traiter au mieux les attentes du client. La connaissance accrue du client permettra de répondre davantage à ses attentes en lui proposant des services plus ciblés. Cela permet de fidéliser le client, enjeu capital sur un marché qui deviendra concurrentiel à partir de 2007.

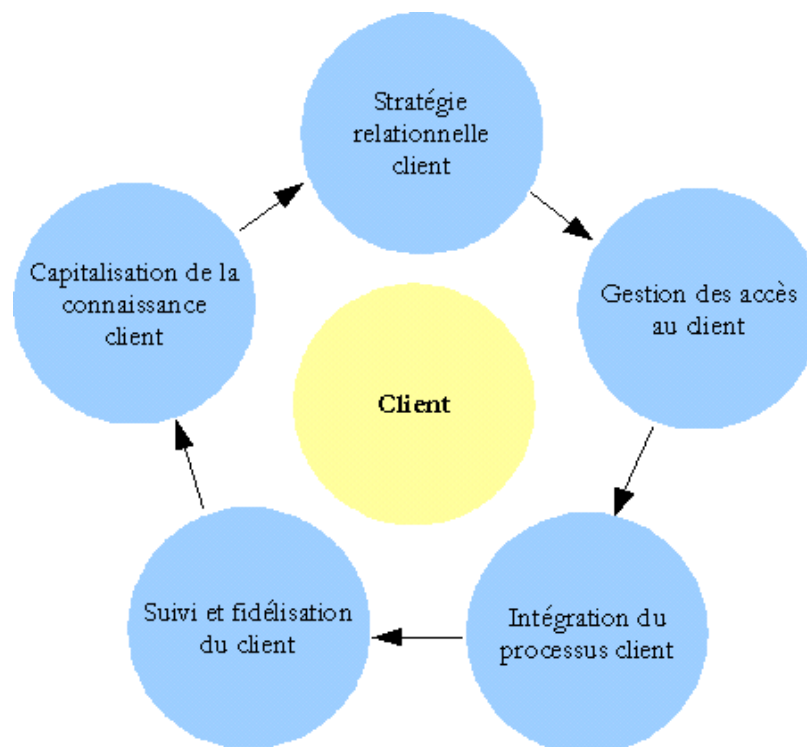


Figure 1 - Dynamique de la GRC

Dans le Conseiller Interactif, nous allons principalement nous intéresser des deux dernières phases de cette dynamique : le suivi et la fidélisation du client et la capitalisation des informations que l'on a de ce dernier.

En prenant un peu de recul et en examinant la GRC chez EDF, elle peut être résumée ainsi :

- 24 millions de clients particuliers,
- 2000 agences,
- des centres d'appels,
- 9 000 conseillers en ligne,
- une présence chez des partenaires,
- deux principaux systèmes d'Informations (Optimia et Impala) capitalisant les informations

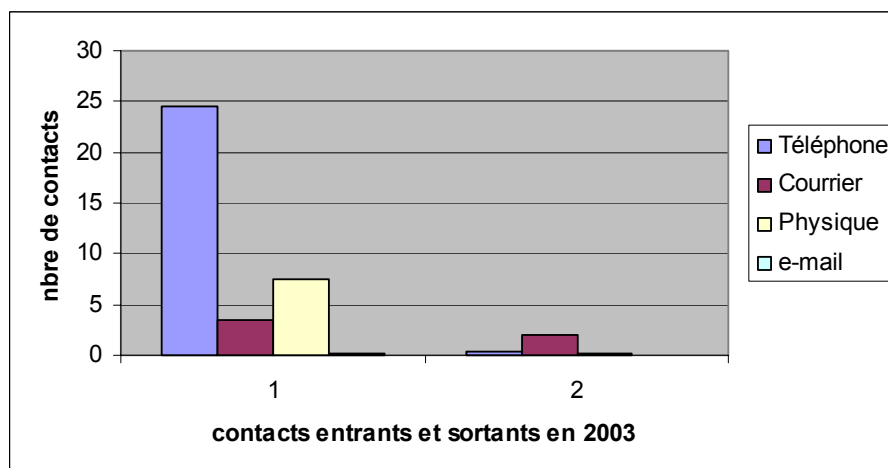
EDF a pour objectif dans le cadre de la GRC d'optimiser la « relation client » en se rapprochant de ce dernier. Différents canaux sont disponibles afin de communiquer avec lui et le fidéliser : le canal physique, le téléphone, le courrier, l'e-mailing, .... Ceux-ci reposent sur diverses plateformes techniques : les centres d'appels et autres SVI<sup>4</sup> pour le canal téléphonique, plateformes informatisées et des conseillers spécialisés pour l'« e-mailing » du canal Web. Le site Internet EDF (Particuliers, Pro, Entreprises et Collectivités Locales) est la pièce maîtresse de ce dernier canal. En effet, l'internaute qui recherche une réponse à une interrogation pourra consulter les informations se trouvant sur le site ou contacter des conseillers par « e-mail ».

## 1. Les canaux traditionnels

Aujourd'hui, EDF dispose d'un large réseau de contacts clients traditionnels : commerciaux, points de vente, catalogues, courriers et centres d'appels.

En 2003, chez EDF, les canaux les plus utilisés étaient :

- En contacts entrants : le téléphone (environ 25 millions d'appels) et le canal physique
- En contacts sortants : le courrier et le téléphone



<sup>4</sup> Serveur Vocal Interactif

Aujourd'hui comme en 2003, le canal téléphonique reste saturé et le sera d'autant plus lorsque les 9000 conseillers en ligne actuels seront répartis entre EDF et Gaz de France. Il faut donc privilégier les autres canaux qui permettront d'assurer une plus grande interactivité avec les clients. L'objectif est d'« humaniser » les autres canaux et de faire avancer le « libre service ». Tout cela permettra l'optimisation des coûts et une meilleure gestion des effectifs.

## 2. Le canal Internet

Internet est un véritable outil d'interaction et de soutien au différent processus marketing, il permet :

- D'accroître sa notoriété avec des coûts faibles,
- Déterminer les marchés cibles et les toucher,
- Tester et accompagner de nouveaux produits ou services,
- Accroître les ventes, réaliser des économies de marché et transformer un prospect en client,
- Diffuser des informations à moindre coût
- Rationaliser les coûts d'exploitation liés à la GRC (canal téléphonique par exemple),
- Mieux répondre aux attentes des clients en développant de l'interactivité en temps réel,
- Fidéliser les meilleurs clients à l'aide de messages d'entreprise afin de réaffirmer les valeurs de la compagnie,
- Proposer un service disponible 24h/24 et 7j/7
- Améliorer la connaissance client en recueillant des informations à travers l'analyse des logs,
- Personnaliser la relation client selon le stade dans lequel il se trouve.

Le canal Internet nous permet de connaître le stade dans lequel se trouve l'internaute dans sa relation avec la marque et ainsi de l'accompagner efficacement d'une phase à une autre :

- Le profil client enregistré implicitement qui semble être la meilleure méthode pour obtenir les informations sur le client et ainsi personnaliser l'interaction sans que l'internaute ait à s'identifier ou à nous donner des informations explicites (en remplissant un questionnaire).
- Les fichiers logs qui permettent de tenir compte du fait qu'un visiteur est déjà venu. Le visiteur récurrent n'est pas ainsi traité comme un « inconnu ». On peut alors masquer certaines informations inadaptées et proposer des informations plus pointues qui lui permettra d'approfondir sa relation avec la marque. On crée ainsi une vraie relation en proposant un site capable d'évoluer et de suivre le visiteur au cours de l'approfondissement de sa relation.
- Les jeux qui sont en ligne sont un autre outil utile à la fois pour recueillir des informations sur les visiteurs mais aussi pour mettre en avant les qualités de vos produits et services ou certaines options qui auraient pu ne pas attirer l'attention des visiteurs.

L'objectif ultime du site Internet est de pousser le visiteur à interagir avec la marque. Une relation de confiance doit alors être établie avec le visiteur, afin qu'il accepte de nous donner plus d'informations le concernant, pour cela il a besoin de trouver un réel gain immédiat. Par exemple : si cet utilisateur fournit des informations le concernant sur ses préférences ou attentes,

le site ne doit alors pas rester le même car il risque de ne faire l'effort de remplir un autre questionnaire.

Afin d'interagir avec la marque, l'outil le plus efficace sur le Web est probablement le « chat ». Il permet de proposer à l'internaute de rentrer en contact avec un conseiller et obtenir une aide en temps réel.

L'utilisation d'Internet est donc essentielle dans la gestion d'une relation multi-canal et la mise en œuvre de solutions supplétives basées sur Internet est nécessaire afin de tirer parti à moindre coût de toutes les richesses qu'il offre.

### 3. Les systèmes d'informations

La GRC est entièrement basée sur l'utilisation des Systèmes d'Information : celle-ci regroupe les outils permettant d'intégrer les clients dans le système d'information de l'entreprise.

EDF dispose d'une multitude de Systèmes d'Informations, je ne vais ici détailler tous ces systèmes. En effet, mon stage de Master en Ingénierie Informatique en avril 2004 consistait à mettre en œuvre un ETL afin de structurer quelques une des ses nombreuses SI dans le but d'effectuer de la fouille de données.

Nous allons principalement nous intéresser aux données récupérées dans le cadre de l'Agence En Ligne (AEL), site Internet mixte EDF/Gaz De France mis à la disposition des clients particuliers depuis plusieurs années. Ce service en ligne propose de nombreux services et agrège de nombreuses informations sur le contrat du client ainsi que sur sa consommation.

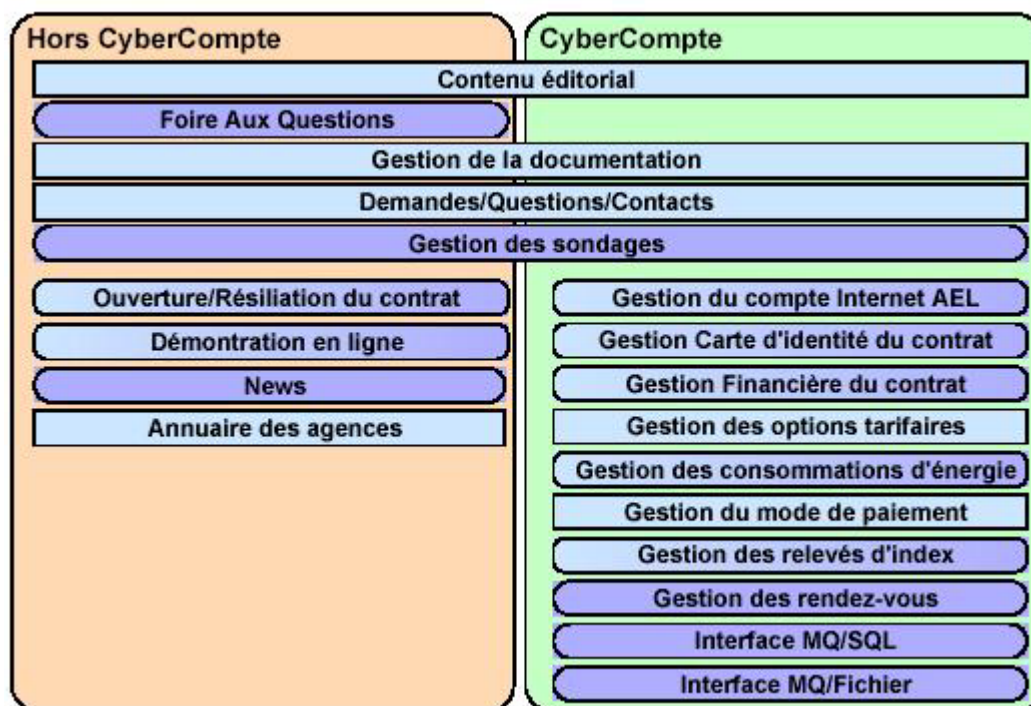


Figure 2 - Architecture de l'Agence En Ligne

Comme nous pouvons le constater en examinant le schéma précédent, l'Agence En ligne contient de nombreuses informations qui pourraient nous être utiles dans le cadre de la GRC.

L'accès à ses informations requiert cependant l'identification du client (connexion par login et mot de passe) et de ce fait le croisement de ces données avec celle d'une autre S.I. est complexe.

La nouvelle génération du Système d'Information EDF verra le jour en 2007, SIMM ou Système d'Information Marché de Masse devrait agglomérer tout les S.I. EDF et proposer les interfaces standardisées permettant d'y accéder.

SIMM incorporera dans sa version simplifiée : gestion de contrats, facturation, recouvrement et gestion de la relation client (réclamation, télévente, ...).

▶ Il serait alors intéressant de pouvoir coupler le système que nous allons mettre en place avec SIMM en utilisant par exemple les Services Web.

## 4. Appui GRC

Afin d'optimiser la mécanique établie via les différents canaux EDF, le projet « Conseiller Interactif » prend forme début 2005, au sein du groupe Solutions et Techniques Innovantes pour La Relation Clientèle. Ce projet fait alors parti d'un autre projet plus conséquent dont le but est l'appui à la Gestion de la Relation Client (Appui GRC).

L'action se situe dans un contexte où la convergence et l'optimisation des coûts des différents canaux de contacts avec la clientèle sont des objectifs prioritaires pour EDF, le département ICAME lance en 2005 une action de recherche innovante dont l'objectif est le développement d'un « Conseiller Interactif sur Internet » de manière à enrichir l'interaction avec le client en étant proactif.

### C. Stratégie et organisation

L'équipe pluridisciplinaire du « Conseiller Interactif » est composée de huit personnes, chacun ayant un rôle particulier et des connaissances métiers spécifiques. J'ai essayé de représenter l'organisation de l'action et les diverses tâches qui nous incombait :

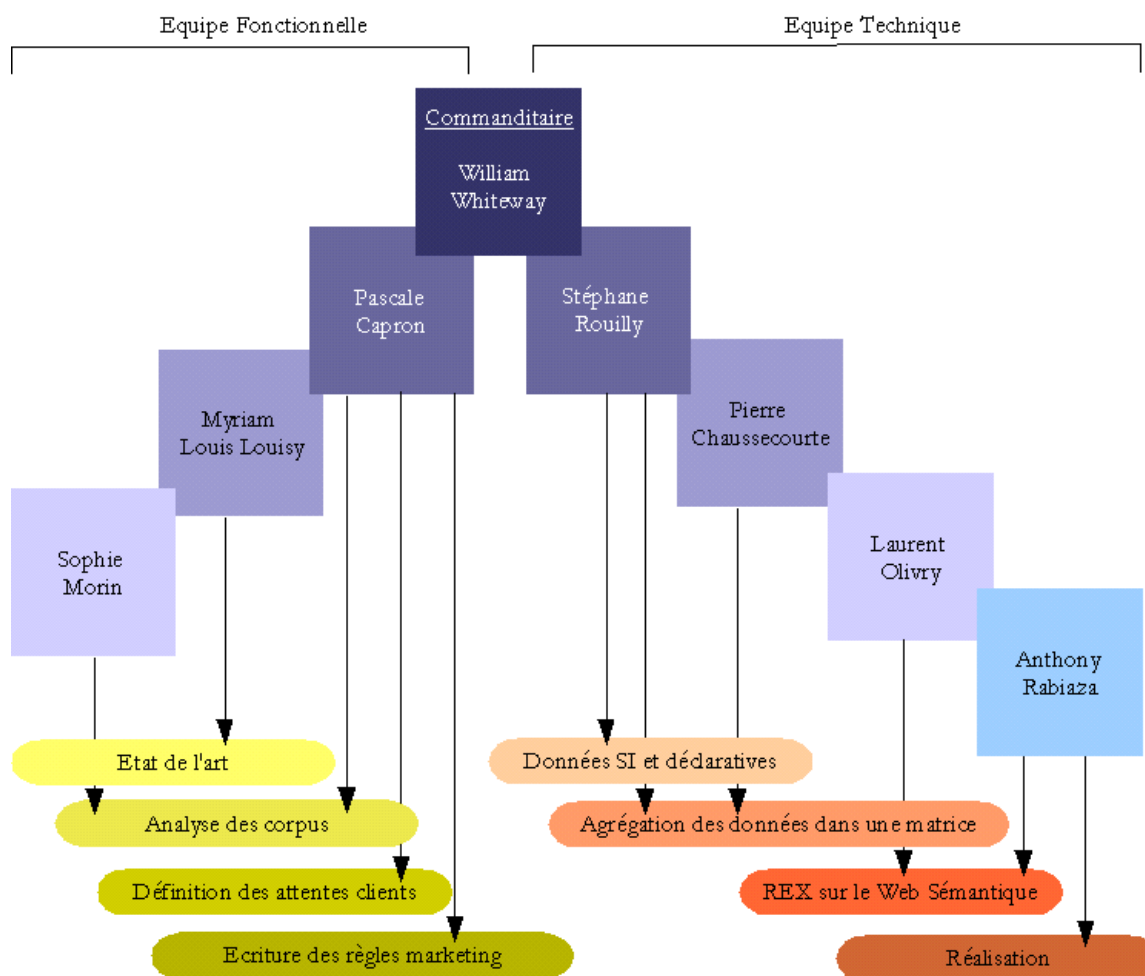


Figure 3 - Organisation de l'action



Voici une brève descriptions des personnes participant au projet :

- **William Whiteway** : commanditaire de l'action, Webmaster du site Particuliers, travaillant à la Division des Particuliers & des Professionnels (DP&P) ;
- **Stéphane Rouilly** : co-responsable de l'action, Ingénieur Chercheur responsable de l'intégration des Nouvelles Technologies sur le Web, travaillant au sein du groupe E76
- **Pascale Capron** : co-responsable de l'action, Ingénieur Chercheur, expertise en processus métier et GRC, travaillant au Groupe E77 - Innovation et Processus de Conception,
- **Pierre Chaussecourte** : responsable de l'intégration (Web Sémantique), groupe E76
- **Myriam Louis Louisy** : ingénieur de recherche en Marketing, du groupe E77,
- **Sophie Morin** : stagiaire de Master en Marketing,
- **Laurent Olivry** : ingénieur de recherche du Groupe SINETICS<sup>5</sup>,
- **Anthony Rabiaza** : Consultant du Groupe Devoteam,

Afin de mener à bien cette mission, l'équipe pluridisciplinaire a établi diverses étapes pour le « Conseiller Interactif », chaque étape donnant lieu à la remise d'un livrable :

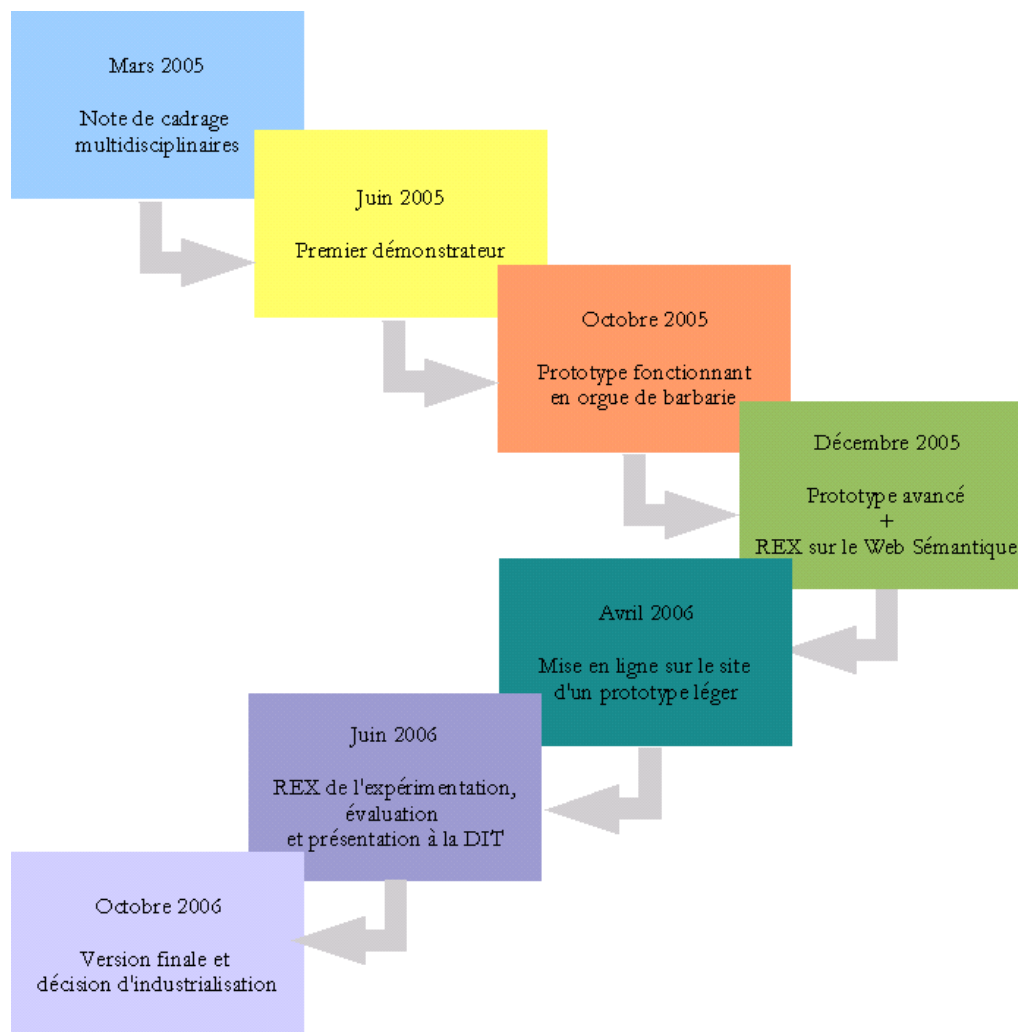


Figure 4 - Livrables

<sup>5</sup> Simulation Neutronique Technologies de l'Information et Calculs Scientifiques

### III. Objectifs du Conseiller Interactif

#### *A. Description de l'action<sup>6</sup>*

L'action « Conseiller Interactif » a pour but d'enrichir l'interaction avec le client sur le Web en étant pro-actif, autrement dit d'automatiser la relation client en anticipant ses demandes.

Pour ce faire, il s'agira de savoir décrypter les freins et les motivations du client en se basant sur son surf, ses déclarations voire sur son profil client issu du Système d'Information EDF, afin de lui proposer soit l'aide, le bon conseil, la bonne offre, le bon outil au meilleur moment et de la façon la plus appropriée, soit l'orienter au plus tôt ou le router vers le canal le mieux adapté à son besoin.

Il s'agira de traiter les attentes de façon automatisée autant que possible et réserver les contacts (téléphone, chat ou mail) à coûts élevés pour EDF aux demandes à forte valeur ajoutée.

Cela devrait contribuer à diminuer les appels entrants des Centres d'appels et permettre aux conseillers de se consacrer aux appels demandant une plus forte expertise (souscription d'offres commerciales...) qu'une simple demande d'information.

Pour que le projet aboutisse, il faut se mettre dans l'idée d'un système automatisé pour permettre à l'internaute d'obtenir des informations rapides sans pour autant déshumaniser la relation client.

Le conseiller interactif permettra alors d'accompagner l'internaute dans sa navigation : il lui proposera la meilleure assistance, au meilleur moment et de la manière la plus appropriée.

---

<sup>6</sup> Compte rendu de réunion : Organisation du projet Conseiller Interactif (20 janvier 2005), Pascale Capron, Myriam Louis Louisy, Pierre Chaussecourte, Stéphane Rouilly, Sophie Morin

## ***B. L'existant : le site Web « Particuliers » et les services interactifs***

Nous allons ici détailler l'espace Particuliers. Espace composé d'informations et de services venant en aide à l'internaute. Ce canal acquiert une grande notoriété auprès des particuliers ; en effet, le site a accueilli 5 millions d'internautes en 2004 et ce nombre de visites est en très forte croissance en 2005 car cette même barre a été atteinte en dès la premier semestre de cette année.

On pourrait segmenter la population des internautes en trois catégories de visiteurs :

- Des clients ayant des projets précis. Ils se rendent sur le site afin de trouver des réponses à des questions spécifiques.
- Des utilisateurs à la recherche de conseils sur le confort électrique et la maîtrise de l'énergie.
- Des internautes naviguant de façon ludique. Ces derniers n'ont pas d'objectif spécifique et parcourent le site Particuliers au grès de leurs humeurs.

### **1. Le site Web « Particuliers »**

Le site Particuliers EDF occupe une position importante au sein du portail EDF. Ce dernier est composé de quatre sites, chacun s'adressant à un segment de la clientèle : le site des « Particuliers », le site des « Professionnels », le site des « Entreprises » et le site des « Collectivités locales ».

Le site Particuliers repose sur un serveur Web Apache connecté à un serveur d'applications. Le rôle du premier est le stockage d'informations statiques et du second la fourniture de services interactifs.

L'architecture succincte du site Particuliers peut être présentée de la manière suivante :

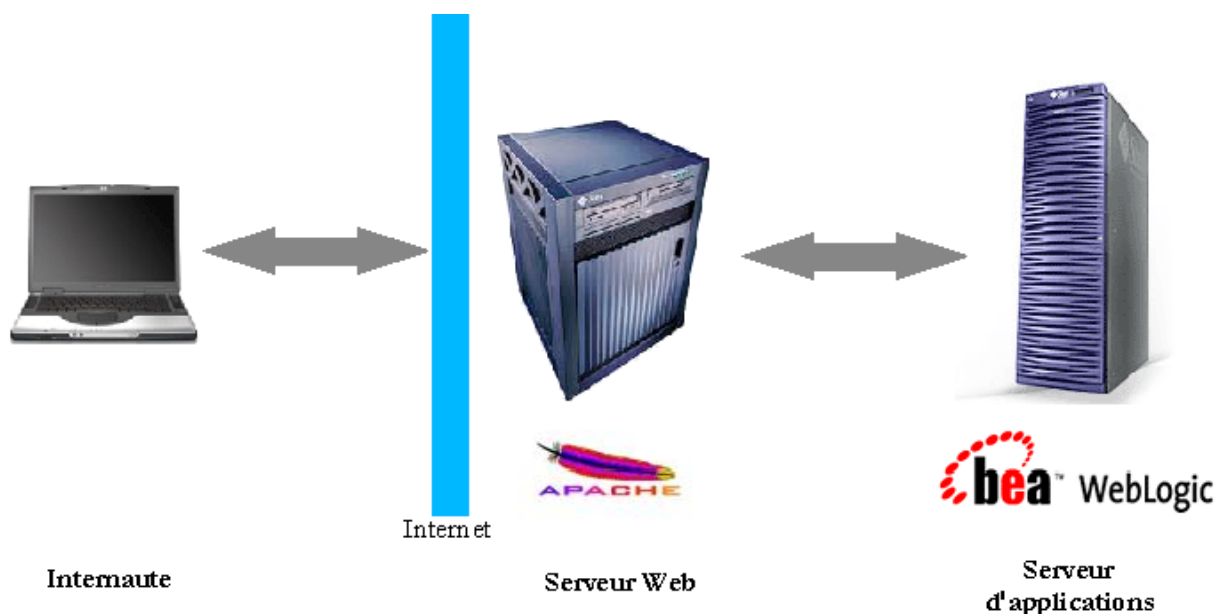


Figure 5 - Architecture générale du site Internet

Les interactions entre l'internaute et le site peuvent être résumés ainsi :

Internaute	Serveur Web	Serveur d'application
L' <b>internaute</b> se connecte au site Web EDF : <a href="http://particuliers.edf.fr">http://particuliers.edf.fr</a> et navigue sur le site.	Le <b>serveur Web</b> de type Apache contient des articles, des brochures, des jeux interactifs et toutes sortes de documents.	Les <b>services interactifs</b> qui permettent à l'internaute de prévoir ses consommations et mieux connaître ses équipements ménagers.

▶ Un grand nombre de documents est alors accessible sur le site Particuliers : il existe plus de 450 pages HTML et 450 documents PDF sur le serveur Web.

## 2. Les services interactifs

Divers services sont proposés à l'internaute aux moyens des « services interactifs » :

- Le **calcul de la facture** (ou CIB), permettant d'estimer la prochaine facture,
- Les **prévisions des dépenses annuelles** (ou CTS) qui permet l'estimation des dépenses annuelles du client,
- Les **coûts de revient des appareils électroménagers** (ou CPA),
- Un **service d'autodiagnostic** qui permet au client de faire le point sur l'ensemble des composants de son logement.

Ces services interactifs sont implémentés sous forme de composants logiciels confiés à un serveur d'application J2EE de type BEA Weblogic.

Les requêtes émanant de l'internaute lors de l'utilisation de ces services transitent cependant par le serveur Web qui les redirige alors vers le serveur d'application : un connecteur Jk est utilisé pour la liaison de ces deux serveurs.

▶ Il est alors possible, en rajoutant des composants à cette architecture, de proposer un « service de conseil interactif » à l'internaute. Ce service utiliserait les données issues de sa navigation : surf et informations déclaratives, voire des données provenant d'un SI.

### 3. Architecture globale à mettre en place

Le « Conseiller Interactif » vient alors se positionner entre l'internaute et le site internet. Son rôle est de prodiguer des conseils à l'internaute, mais avant cela, il doit construire le profil client au fur et à mesure de la relation que l'on entretient avec lui depuis sa connexion.

L'apparition est alors conditionnée par le profil client et l'évaluation de la pertinence du déclenchement (et éventuellement des connaissances provenant du SI) :

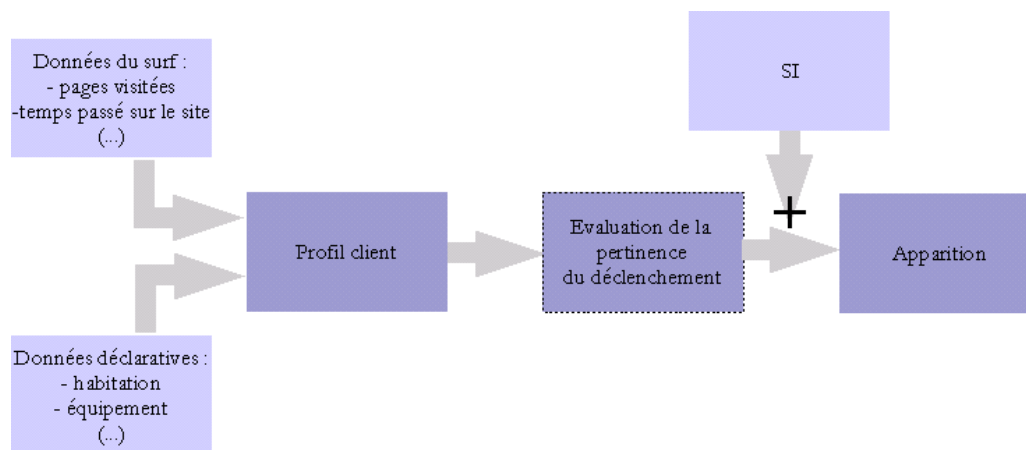
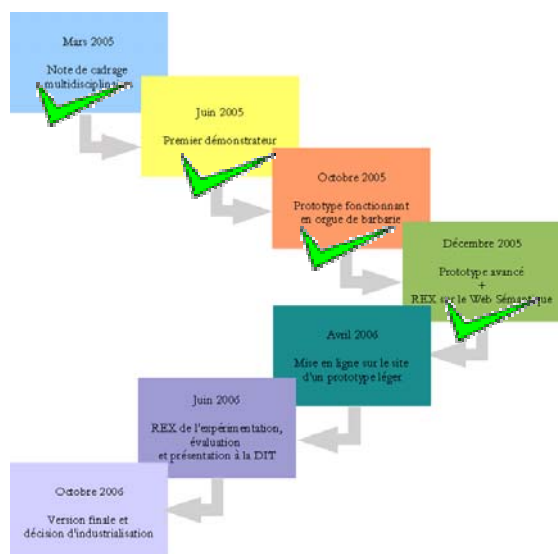


Figure 6 - Architecture à mettre en place

### C. Synthèse de ce qui a été fait

Depuis la note de cadrage (Mars 2005), neuf mois se sont écoulés et tous les objectifs fixés dans ce laps de temps ont été atteints. En effet, un premier démonstrateur a été livré en juin 2005, le prototype en orgue de barbarie délivré en octobre et présenté lors du Coff'icame<sup>7</sup>.



Le projet entre le mois d'octobre et de décembre a pris une certaine avance et de ce fait le prototype avancé gèrera beaucoup plus d'internautes que dans les premières estimations (quatre fois plus : 20 internautes en simultanément au lieu de 5).

Le REX sur le Web sémantique est en cours de rédaction et sera remis avant la fin de l'année.

Les objectifs pour l'année 2006 sont la continuelle amélioration de la plateforme afin qu'en avril 2006 un prototype léger puisse être mis en ligne, la modularisation de la plateforme afin que l'on puisse utiliser les outils et les connaissances du « Conseiller Interactif » sur d'autres projets<sup>8</sup>.

<sup>7</sup> Réunion du groupe ICAME le 13 octobre 2005

<sup>8</sup> voir Perspectives

## IV. Les technologies

Plusieurs solutions techniques existent pour la gestion de l'interaction avec les clients sur le Web. Nous allons d'abord vous présenter des solutions de CRM clés en main puis un panel d'outils de plus bas niveaux issus du monde du Web Sémantique.

### ***A. Les solutions clé en main***

Les outils de e-CRM se déclinent autour de trois axes d'activités : marketing, commercial et service client. Les fournisseurs de solutions pour la gestion de la relation client proposent alors des solutions modulaires centrées autour d'une base de données clientèle. Elles permettent :

- L'automatisation des activités marketing de l'entreprise (EMA<sup>9</sup>), ceci afin automatiser toutes les tâches relatives à la GRC. Divers modules se chargent alors de trois types de tâches : *événementielles* (ex : courriers de bienvenue), *programmées* (ex : relance téléphonique) ou *conditionnelles* (c'est des tâches événementielles s'aidant de « business rules<sup>10</sup> »).
- L'automatisation de la prise en charge commerciale du client (SFA<sup>11</sup>). Cette prise en charge permet de mieux appréhender les besoins du client selon les informations que l'on possède de lui, l'objectif étant d'être réactif dans des délais très courts.
- Le support du service client (CSS<sup>12</sup>) qui utilise principalement la base de données clientèle. Cette partie concerne principalement le service après-vente de l'entreprise.
- La modélisation et la segmentation de la clientèle via un module statistique.

Nous allons maintenant voir un peu plus en détail les outils d'e-CRM afin d'en extraire les informations qui pourront nous servir pour notre plateforme et nous détaillerons ensuite le « cœur » des outils d'e-CRM : les moteurs de règles.

---

<sup>9</sup> Enterprise Marketing Automation

<sup>10</sup> Règles marketing

<sup>11</sup> Sales Force Automation

<sup>12</sup> Client Service Support

## 1. Les outils d'e-CRM<sup>13</sup> clés en main

Les outils que nous vous présentons ici sont les solutions techniques les mieux adaptées et les plus utilisées pour l'interaction avec les clients sur le Web.

### a) Vignette<sup>14</sup>



Vignette permet aux entreprises de construire, de déployer et d'optimiser rapidement des applications Web. Vignette commercialise alors des solutions permettant à l'entreprise d'améliorer considérablement l'expérience client sur leur site.

Quatre catégories de services d'applications sont comprises dans l'offre technologique de Vignette :

- Intégration,
- Information,
- Interaction,
- Management

Ces solutions devraient alors permettre de délivrer la bonne information, à la bonne personne, au bon moment.

Pour la gestion des campagnes Marketing et des initiatives d'écoute client, Vignette propose **Vignette Dialog**. Ce dernier permet de :

- Orienter les clients au travers du cycle de vente sur le site Web avec des dialogues personnalisés délivrés par e-mails, courrier physique ou un autre mode de communication,
- Obtenir des données sur les besoins individuels,
- Exécuter des actions pertinentes déclenchées automatiquement à partir de règles marketing.

---

<sup>13</sup> Electronic Customer Relationship Management

<sup>14</sup> <http://www.vignette.com/contentmanagement/0,2097,1-1-1928-4149-1968-4161,00.html>

Vignette Dialog comporte une multitude de composants dédiés, connectant les services clientèles au client via plusieurs liaisons :

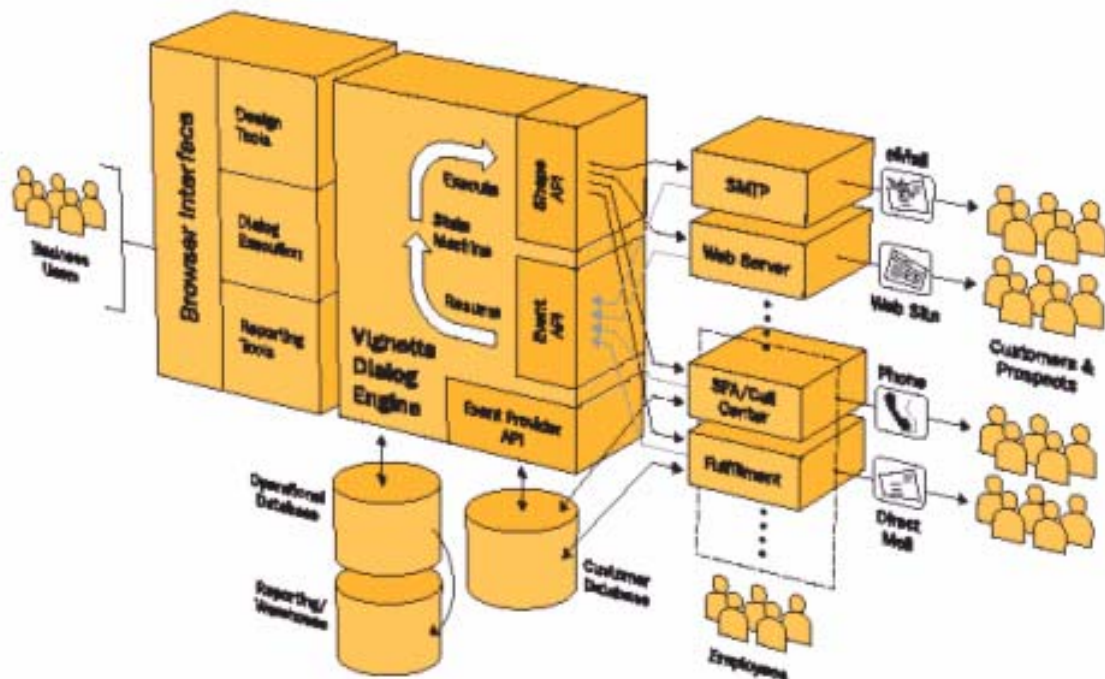


Figure 7 - Architecture de Vignette Dialog

<http://www.vignette.com>

Une autre solution de Vignette, **Customer Self-Service Solutions**, existe également. Il s'agit d'une solution self-service qui permet au client de consulter ses informations personnelles et gérer son compte. Cette solution permet de mieux cadrer les offres en utilisant les préférences clients, ses historiques, le tout afin de réduire la charge des call-center et l'amélioration de la satisfaction client grâce à une communication personnalisée.

Les avantages d'une implémentation réussie de la solution self-service :

- Elle donne au client l'accès à des informations ciblées,
- Elle permet au personnel du service client de localiser rapidement et facilement les documents adaptés à chaque profil et d'accéder aux historiques.
- Elle permet de créer un environnement favorable pour la vente de produits additionnels à des clients existants.



## b) Broadvision<sup>15</sup>

Broadvision est un fournisseur d'applications Web Self-service personnalisables. Cette entreprise est reconnue comme pionnière en matière de personnalisation et d'interactions client via le Web. Broadvision propose One-To-One Portal : c'est une application pour le déploiement de portails personnalisés Business-to-Employees, Business-to-Business ou Business-to-Customer.

Quatre secteurs utilisent cette solution : le secteur de télécommunications, le « retail », la fabrication et distribution. La première et les secondes utilisent des applications Web Self-Service de Broadvision afin de mettre en place et personnaliser des initiatives on-line afin de réduire les coûts des call-center, les autres utilisent la solution de Broadvision afin de concrétiser leurs stratégies on-line.

La personnalisation et les interactions sont facilitées par le portail One-to-One. Ainsi le nouveau portail de Hewlett-Packard permet aux clients de réduire leurs coûts de transaction ; il offre aux utilisateurs un accès à ses informations sur les produits, sur leur compte ainsi qu'un site personnalisable avec des options d'opt-in<sup>16</sup> et d'opt-out<sup>17</sup>.

## c) E-deal Automation<sup>18</sup>

Automation est un système entièrement automatisé qui utilise la connaissance client, les règles de gestion marketing et la démarche commerciale de l'entreprise pour créer et développer une relation commerciale personnalisée avec chacun des clients pris individuellement. Il permet de proposer à chaque client l'offre personnalisée qu'il attend au moment où il l'attend.

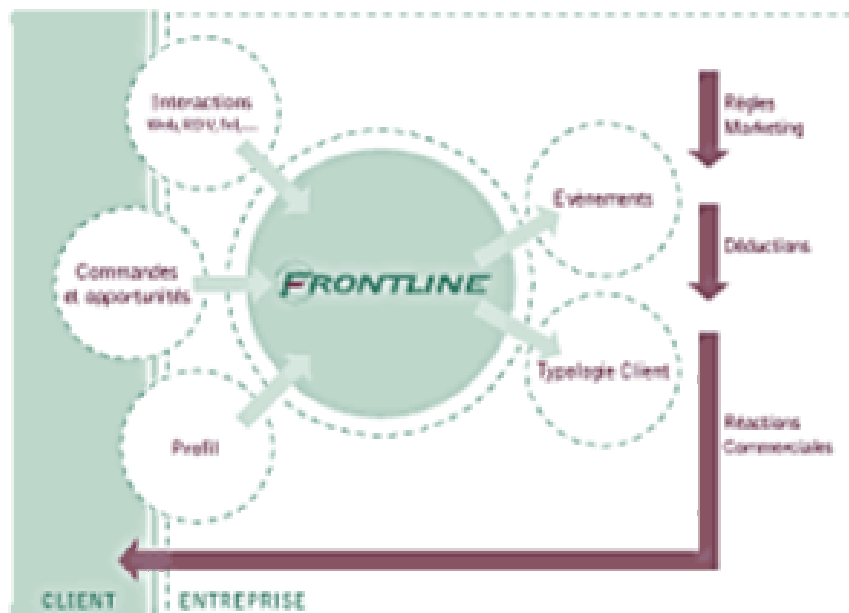


Figure 8 - E-deal Automation

<http://www.e-deal.com/software/automation.jsp>

<sup>15</sup> <http://www.broadvision.com>

<sup>16</sup> Accord de l'internaute explicite pour une collecte de données personnelles

<sup>17</sup> Les données sont automatiquement collectées car on considère l'accord de l'internaute acquis par défaut

<sup>18</sup> <http://www.e-deal.com/software/automation.jsp>

#### **d) Les solutions Liveperson<sup>19</sup>**

Liveperson est un fournisseur de solutions d'hébergement pour la gestion des interactions client : l'entreprise permet alors d'identifier et d'attirer le bon client de façon pro-active au meilleur moment en utilisant le bon canal de communication. Timpani SB, est une solution de communication multi-canal facilitant la vente en temps réel, optimisant l'efficacité du service client et facilitant les remontées d'informations vers le marketing. Les outils fournis par Timpani supportent l'interaction avec le client en ligne : chat, email et self-service.

#### **e) Les solutions Crafty Syntax Live Help<sup>20</sup>**

Contrairement aux solutions précédentes, Live Help est un projet OpenSource supportant le « chat » en ligne et permet aux opérateurs de sites Web de suivre les visiteurs pendant leur surf et leur proposer de façon pro-active des sessions de « chat ».

## **2. Fonctionnalités à retenir pour notre plateforme**

En recoupant les besoins du « Conseiller Interactif » et les fonctionnalités des solutions d'e-CRM, nous pouvons retenir les points importants pour notre plateforme, celle-ci doit donc :

- Se fondre au site Web EDF fonctionnel,
- Avoir des capacités d'intégration avec l'architecture existante (les services interactifs),
- Se déployer sans aucune installation sur le poste du client (celui de l'internaute),
- Personnaliser et historier automatiquement les interactions client,
- Etre modulaire et extensible afin que le rajout de fonctionnalités soit aisé.

► L'étude des fonctionnalités présentes dans les différentes solutions d'e-CRM permet d'établir une architecture pour le « Conseiller Interactif ». Celle-ci s'intégrera à l'existant, sera modulaire et extensible.

---

<sup>19</sup> <http://www.liveperson.com/sb>

<sup>20</sup> <http://www.craftysyntax.com>

### 3. Les BRMS<sup>21</sup> : moteurs de règles

Le moteur de règles est au cœur des solutions de GRC. Il s'appuie sur un référentiel central de règles susceptible de servir les différentes applications. On peut s'y référer dans la logique services Web ou aussi embarquer des moteurs à chaque étape. Dans le premier cas, on simplifiera les problèmes de cohérence des règles. Dans le second, on sera plus performant. Tout dépend du volume de consultation. Deux types de processus sont externalisés. D'un côté, l'enchaînement des étapes, l'acheminement d'un dossier, par exemple, comme dans un Workflow. De l'autre, la réglementation proprement dite.

#### a) Qu'est-ce qu'un moteur de règles ?

Selon l'Entreprise Daumas<sup>22</sup>, « Les règles métiers permettent de décrire précisément les critères de décision et les actions à mener pour un processus donné. Ces règles, qu'elles soient formalisées ou non, existent dans chaque cœur de métier. Elles capitalisent la connaissance, le savoir-faire d'une entreprise et traduisent sa stratégie »

Le moteur de règles permet alors d'externaliser les règles métiers dans des fichiers externes : des sources formalisées (ex : fichiers XML, EJB, WebServices, ...) et effectuer des décisions et actions selon les règles applicables.

Le fonctionnement d'un moteur de règles peut-être illustré de la manière suivante :

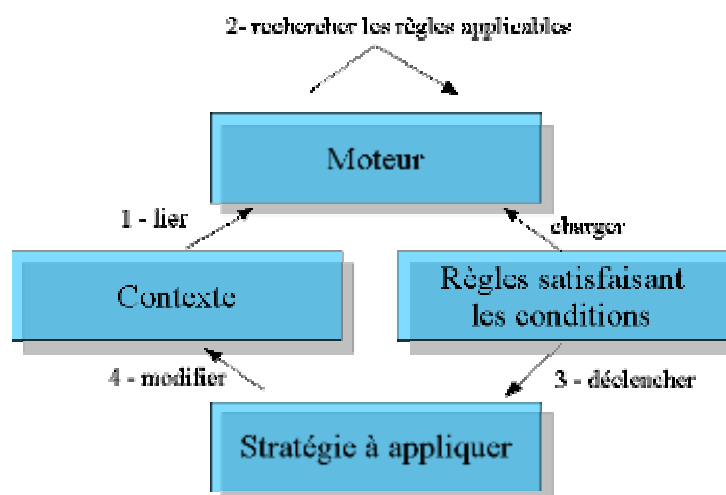


Figure 9 - Un moteur de règles : comment ça marche ?

<sup>21</sup> Business Rule Management System

<sup>22</sup> <http://www.daumas-autheman.com>

Les grands éditeurs de logiciels se tournent à présent vers ce domaine et développent des outils capables de gérer le Workflow avec des moteurs de règles, permettant ainsi l'automatisation des processus métiers.

Voici un tableau présentant diverses solutions :

TOUS LES ÉDITEURS Y VIENNENT PEU À PEU		
Editeur	Produit	Commentaires
<b>IBM</b>	WebSphere Business Integration Workbench	Interface graphique. Possibilité de simuler l'impact de la modification d'une règle. Plusieurs moteurs de règles (MQ Workflow, Business Integration Message Broker, Business Integration Event Broker, Interchange Server).
<b>Pegasystems</b>	Pegarules 4.0	Règles paramétrables par les experts métier. Gestion des versions de règles et changement de version à chaud. Vérification automatique de la syntaxe, etc.
<b>Savvion</b>	BPM Studio et Rule Editor	Outil d'édition et moteur d'exécution des règles totalement indépendant de BPM Server. En revanche, aucun connecteur vers des moteurs de règles externes. Moteur natif.
<b>Staffware</b>	Process Suite 10	Règles paramétrables sans développement par les experts métier. Simulation.
<b>W4</b>	W4 Engine 4.1 et W4 Studio	Règles paramétrables par les experts métier. Axé sur la réutilisation des définitions de workflow (W4 Library). Moteur natif.

Figure 10 - Quelques moteurs de règles

<http://www.01net.com/article/256828.html>

### b) JRules<sup>23</sup> : « Le » moteur de règles



JRules est un moteur de règles écrit en Java qui concurrence les outils de BPM<sup>24</sup>. Ce dernier comprend un éditeur de Workflow, Ruleflow, qui permet de définir et de représenter, sous la forme de schéma UML d'activité, l'enchaînement des actions impliqué dans les règles métiers. Des tables de décision permettent de regrouper en tableaux des règles de nature similaire (ex : celles liées à une politique de tarification). Le tout afin de simplifier la représentation du processus.

L'augmentation prévisible du nombre de règles est évidemment prévue, le moteur assure donc une vérification de la cohérence des règles. Question ouverture, il supporte des standards des services Web, ce qui permet la publication de règles en tant que services et leurs invocations via leurs descriptions WSDL.

<sup>23</sup> <http://www.ilog.fr/products/jrules/>

<sup>24</sup> Business Process Management

JRules comprend trois principaux modules pour chaque type d'utilisateur :

*Business Rule Development* (Constructeur, Repository, Editeur et Vérificateur)

→ pour le développeur qui assure le déploiement des règles

*Business Rule Management* (Langage de règles et outils de gestion)

→ pour l'analyste en business qui crée, édite ou supprime des règles

*Business Rule Execution* (Déploiement, intégration, ...)

→ pour le spécialiste métier, celui-ci rédige ou rejette les règles

Le langage d'écriture des règles se rapproche du langage naturel :

**If the age of the customer is greater than 18 And ... Then ...**

Au niveau architecture, JRules peut être aisément intégré à du J2EE, .Net ou en utilisant les Web Services, voici un schéma<sup>25</sup> présentant l'intégration du moteur JRules avec un serveur d'application J2EE :

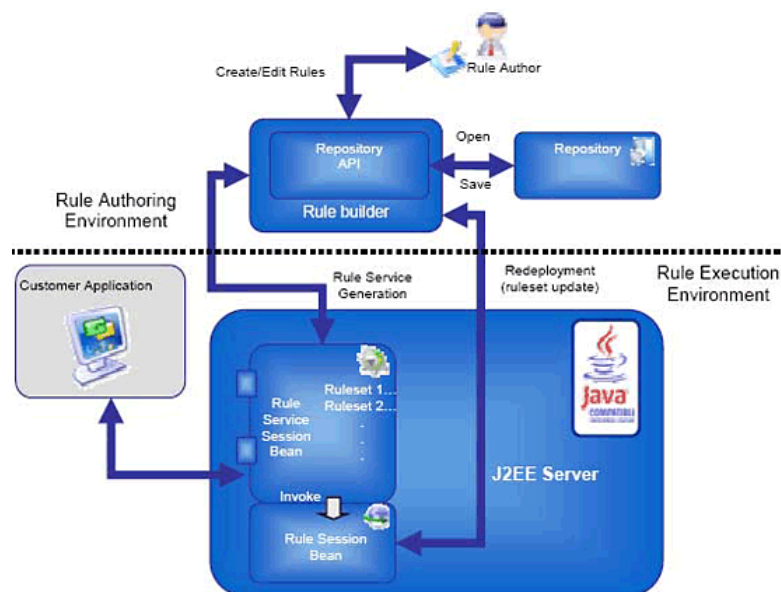


Figure 11 - Jrules + J2EE

<http://www.ilog.fr/products/jrules/>

<sup>25</sup> [http://dev2dev.bea.com/pub/a/2005/04/business\\_rules\\_engines.html](http://dev2dev.bea.com/pub/a/2005/04/business_rules_engines.html)

### c) Les moteurs émergents : JESS<sup>26</sup> et Drools<sup>27</sup>

JRules n'est pas le seul moteur de règles existant sur le marché, deux autres outils sont également disponibles :

**JESS** est un moteur de règles écrit en Java disponible gratuitement pour les recherches dans un domaine académique. Il utilise l'algorithme de type « pattern matching » très gourmand en ressources mais rapide.

**Drools** à l'instar de Jess est écrit en Java et est un moteur utilisant l'algorithme de RETE (=réseau en latin, c'est un algorithme incrémental d'unification évitant l'itération) Ses principaux avantages sont la simplicité de son interface graphique et son fonctionnement sous divers plateformes comme nous le montre le schéma suivant :

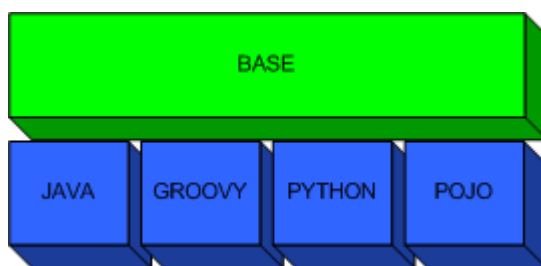


Figure 12 - Drools

<http://drools.org/>

### d) Comparatif

Voici un tableau résumant les fonctionnalités des trois moteurs de règles précédents avec leurs avantages et inconvénients :

Outils	Lecture de règles	Points forts	Points faibles
Ilog JRules	Lecture séquentielle des règles Algorithme de RETE	Rapide et supporte un très grand nombre de règles sans altération des performances Compatibilité Java, .Net et Web Services	Prix (à partir de \$12'000) Limitation induit par le langage « naturel »
Sandia Jess	Algorithme utilisant les « Pattern Matching »	Gratuit Assez rapide	Lourdeur
Codehaus Drools	Algorithme RETE-OO (orienté objet)	Compatibilité (Java, Python, ...)	Complexité d'utilisation

<sup>26</sup> <http://herzberg.ca.sandia.gov/jess/>

<sup>27</sup> <http://drools.org/>

### e) JSR 94<sup>28</sup> : Java Rule Engine API

JSR 94 sont les spécifications pour les moteurs de règles Java. De grands industriels ainsi que des éditeurs (ATG, Fujitsu, IBM, ILOG, Novell, Oracle ou encore Sun) ont contribué à cet effort de standardisation.

L'objectif de JSR 94, dont JRules et Drools respectent les spécifications, est d'être le langage de règles universel de tous les moteurs de Workflow. Les règles sont écrites en XML et utilisent une syntaxe proche du Java :

```
<rule-execution-set>
  <name>RES_0</name>
  <description>RL Text-based Rule Execution Set 0</description>
  <rule-source>
    <rl-text>
      ruleset rs0 {
        class rlc0 { int id; }
        rule r0 {
          if fact rlc0 { println("&quot;rlc0 with id = &quot; + rlc0.id); }
        }
      }
    </rl-text>
  </rule-source>
  <ruleset-stack> <ruleset-name>rs0</ruleset-name> /ruleset-stack>
</rule-execution-set>
```

---

<sup>28</sup> <http://www.jcp.org/en/jsr/detail?id=94>

## ***B. Le Web Sémantique***

Nous allons ici vous présenter le panel technologique du Web Sémantique : de la même manière que les différents protocoles de couches OSI, plusieurs standards W3C s'empilent afin de fournir les services nécessaires à la formalisation des informations et à l'élaboration de plateforme pour les données sur le Web.

### **1. Le XML<sup>29</sup>**



XML (ou eXtended Markup Language) est le langage à balises étendus. C'est le standard du W3C pour la mise en forme de documents. XML a été conçu pour des documents simples ou complexes et s'appuie sur cinq grands principes simples et clairs :

- Lisibilité à la fois par les machines et par les utilisateurs,
- Définition sans ambiguïté du contenu d'un document,
- Définition sans ambiguïté de la structure d'un document,
- Séparation entre documents et relations entre documents,
- Séparation entre documents et relations entre documents,
- Séparation entre structure du document et présentation du document.

XML a été mis au point par le XML Working Group sous l'égide du World Wide Web Consortium (W3C<sup>30</sup>, l'organisme chargé de standardiser les évolutions du Web) dès 1996. Depuis le 10 février 1998, les spécifications XML 1.0 ont été reconnues comme recommandations par le W3C, ce qui en fait un langage reconnu. (Tous les documents liés à la norme XML sont consultables et téléchargeables sur le site Web du W3C).

Dans le monde du Web Sémantique, XML apporte syntaxe, règles et documents structurés. Technologie de base, XML fournit un ensemble de règles pour la création de vocabulaires qui structurent à la fois les documents et les données sur le Web. XML offre des règles de syntaxe très claires. XML est une syntaxe puissante et souple pour les documents structurés, mais n'impose aucune contrainte sémantique sur la signification de ces documents.

---

<sup>29</sup> <http://www.w3c.org/XML/>

<sup>30</sup> <http://www.w3c.org/>



XML fournit alors une base sur laquelle les développeurs peuvent ériger des applications. Afin de formaliser des contraintes, on utilise les DTD (Document Type Definition) ou une version plus élaborée : XML Schema. Il explicite la structure d'un fichier XML. Sa principale raison d'être est de décrire une structure précise et permet ainsi la validation d'un fichier XML par rapport à cette description.

► XML apporte alors syntaxe, règles et structuration des documents.

## 2. RDF<sup>31</sup>



RDF (ou Resource Description Framework) est un standard permettant la mise en place de descriptions de ressources simples. XML est à la syntaxe, ce que RDF est à la sémantique : un ensemble clair de règles permettant des informations descriptives simples. RDF est intégré dans une variété d'applications comme :

- catalogues de bibliothèques,
- annuaires internationaux,
- syndication et agrégation de nouvelles, de logiciels et de contenus,
- collections personnelles de musiques, de photos et d'évènements.

Dans chacun de ces cas d'utilisation, XML peut être utilisé comme une syntaxe interchangeable. Les spécifications RDF nous fournissent alors un environnement de travail puissant pour l'échange et la capitalisation de connaissances sur le Web.

Selon Brian McBride, responsable du groupe de travail RDF Core du W3C : « RDF représente une étape fondamentale dans la montée en puissance du Web. Nous allons enfin voir des applications qui combinent les informations provenant de sources multiples représentées en RDF, ce qui était totalement impensable jusqu'à présent ... ». En effet, ce groupe de travail du Consortium W3C, a permis aux spécifications RDF de se placer comme base du Web sémantique.

Afin de définir la structure des descriptions RDF, le W3C a défini son langage de schéma : RDFS<sup>32</sup> (RDF Schema), celui-ci nous permet ensuite de décrire le contenu du RDF et de combiner ces descriptions en un seul schéma.

► RDF nous apporte alors une infrastructure de données pour le Web.

---

<sup>31</sup> <http://www.w3.org/RDF/>

<sup>32</sup> <http://www.w3.org/RDFS/>

### 3. OWL<sup>33</sup>



L'objectif de OWL (ou Ontology Web Language) est comme son nom l'indique le développement de vocabulaires spécifiques à un sujet - ou un domaine. C'est le rôle d'une ontologie. Une ontologie définit les termes utilisés pour décrire et représenter un champ d'expertise. Les ontologies sont utilisées par les personnes, les bases de données, et les applications, qui ont besoin de partager des informations relatives à un domaine bien spécifique comme, la médecine, la fabrication d'outils, l'immobilier, la réparation d'automobiles, la gestion de finances, etc. Les ontologies associent les concepts de base d'un domaine précis et les relations entre ces concepts, tout cela d'une manière compréhensible par les machines. Elles encodent la connaissance d'un domaine particulier ainsi que les connaissances qui recouvrent d'autres domaines, ce qui permet de rendre les connaissances réutilisables.

OWL - langage d'ontologies Web - est un langage permettant de définir des ontologies Web structurées, et ce faisant, autorise une intégration plus riche et garantit l'interopérabilité des données au travers des frontières applicatives. Les premiers langages utilisés pour le développement d'outils et d'ontologies pour des communautés d'utilisateurs spécifiques (particulièrement en sciences et dans des applications d'e-commerce spécifiques à certaines entreprises) n'ont pas été définis pour être compatibles avec l'architecture du Web en général, et du Web sémantique en particulier.

OWL répare ce manque en utilisant à la fois les URIs pour nommer, et la fonctionnalité fournie par RDF pour créer des liens. Ainsi, les ontologies Web possèdent les avantages suivants :

- Capacité d'être distribuées au travers de nombreux systèmes,
- Mise à l'échelle pour les besoins du Web,
- Compatibles avec les standards Web pour l'accessibilité et l'internationalisation,
- Ouvertes et extensibles.

OWL s'appuie sur un modèle et un schéma RDF pour ajouter plus de vocabulaire dans la description de propriétés et de classes, comme par exemple : les relations entre classes (i.e. la séparation), la cardinalité (i.e. « un et un seul »), l'égalité, une typographie plus riche des propriétés, des caractéristiques de propriétés (i.e., la symétrie), et des classes énumératives.

« OWL constitue une avancée importante dans la représentation et l'organisation des connaissances disponibles sur le Web. OWL réussit à établir un bon équilibre entre les besoins de l'industrie pour un langage prenant en compte leurs cas d'utilisation du Web, et les restrictions imposées par les connaissances et les expériences de la recherche dans le développement d'un langage d'ontologies, » expliquent Jim Hendler et Guus Schreiber, co-animateurs du groupe de

---

<sup>33</sup> <http://www.w3.org/OWL/>

travail Web Ontology du W3C. « La cinquantaine de participants du groupe de travail ont conçu avec brio un langage qui souscrit aux préoccupations communes, et qui soit épaulé aussi bien par les académiques que par les professionnels. »

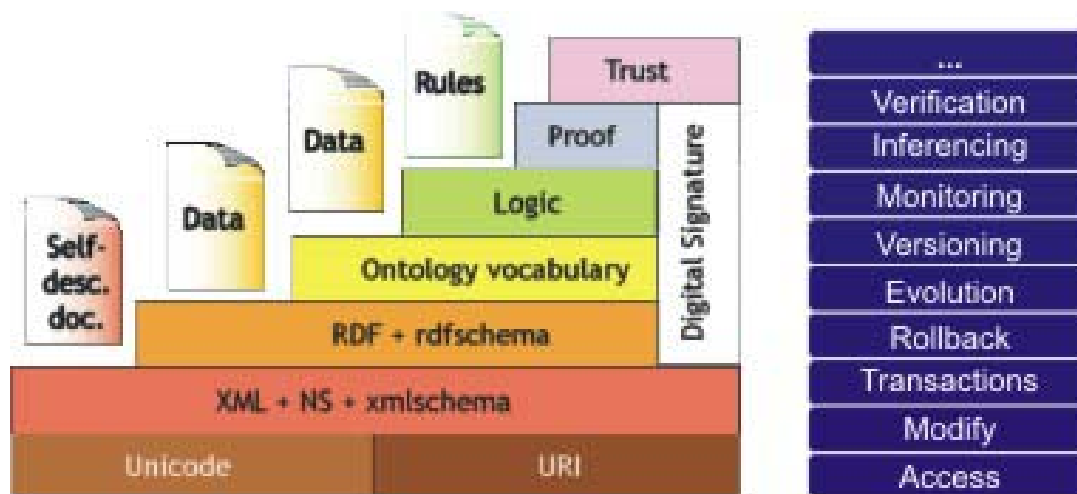
▶ OWL fournit des ontologies opérationnelles pour le Web.

Si on s'intéresse au niveau technique, le langage OWL est dédié aux définitions de classes et de types de propriétés, et donc à la définition d'ontologies. Inspiré des logiques de descriptions (et successeur de DAML+OIL), il fournit un grand nombre de constructeurs permettant d'exprimer de façon très fine les propriétés des classes définies. La rançon de cette expressivité est l'indécidabilité du langage obtenu en considérant l'ensemble de ces constructeurs. Voilà pourquoi OWL est fractionné en trois langages distincts :

- OWL LITE ne contient qu'un sous-ensemble réduit des constructeurs disponibles, mais son utilisation assure que la comparaison de types pourra être calculée (un problème de NP, donc «simple» en représentation de connaissances),
- OWL DL contient l'ensemble des constructeurs, mais avec des contraintes particulières sur leur utilisation qui assurent la décidabilité de la comparaison de types. Par contre, la grande complexité de ce langage (un de ses fragments est P-SPACE-complet) semble rendre nécessaire une approche heuristique,
- OWL FULL, sans aucune contrainte, pour lequel le problème de comparaison de types est vraisemblablement indécidable.

#### 4. Couches du Web Sémantique

Voici la manière dont nous juxtaposeront les diverses couches du Web Sémantique et les fonctionnalités qui nous seront offertes :



Les deux dernières couches sont ambitieuses et devraient apporter au Web Sémantique des notions de preuve et de confiance. Nous ne les utilisons pas dans le prototype du fait qu'elles ne seront disponibles que dans quelques années.

## 5. Avantages « théoriques » du WS

Le Web Sémantique est décrit généralement comme un Web destiné aux machines. Il s'agit pourtant plus d'une infrastructure que d'une application (ou d'un langage). Plusieurs avantages peuvent nous être apportés par ce Web « compréhensible » par les machines :

- L'automatisation de nombreuses tâches fondées sur le contenu (recherche de ressources ayant un contenu particulier, comparaison de ressources par les informations contenues. Le Web sémantique résoudrait alors les problèmes de recherches d'informations sur la toile.
- L'automatisation de tâches liées à la mise en relation de ce contenu comme l'articulation de réponses fournies par plusieurs ressources.
- La description de ressources informatiques et de services selon divers critères (conditions d'activations, résultat, qualité, ...). Ceci devrait alors permettre d'effectuer des recherches et de se connecter à ces ressources.

La véritable « valeur ajoutée » du Web sémantique réside principalement dans l'articulation d'éléments dispersés et la composition de services. L'interopérabilité doit être assurée entre les différentes ressources.

On peut retenir quatre propriétés importantes du Web Sémantique, il doit être :

- **Formalisé** : le Web informel est déjà disponible, c'est la formalisation qui fait le Web sémantique. Bien entendu, il faut savoir lier le formel à l'informel [Euzenat 1996].
- **Ouvert** : le Web ne peut pas être sémantique dans une organisation et syntaxique à l'extérieur. Ce qui est important est que ce qui peut circuler, ce qui peut être appréhendé soit formalisé. C'est à cette condition qu'il sera possible aux machines d'exploiter le Web sémantique et, en particulier, de mettre en relation les ressources.
- **Interopérable** : il est nécessaire de communiquer dans un langage formalisé et de pouvoir manipuler correctement les connaissances qui y sont stockées.
- **Coopératif** : le Web est un ensemble de ressources partagées, la connaissance est alors répartie et doit se transmettre de la même manière que dans les réseaux « pair à pair ».

## 6. Les outils

Nous allons, ici, montrer les différents outils venant nous accompagner dans l'élaboration de solutions basées sur le Web sémantique : Framework, éditeurs, outils de visualisation et autres raisonneurs.

### a) Les API RDF/OWL

Nombreuses sont les boîtes à outils permettant de gérer le RDF et/ou de l'OWL. Elles sont implémentées sous diverses plateformes et divers langages : C, Java, JavaScript, Lisp, .Net, PHP Prolog ou encore Ruby.

Nous allons vous présenter un panorama des solutions existantes sous forme de comparatif ; cela nous permettra de voir leurs avantages respectifs et de sélectionner « notre » boîte à outils.

#### (1) *Tableau<sup>34</sup> comparatif des solutions*

Pour commencer ce comparatif, voici les API disponibles en **langage C** dont le principal avantage réside dans leur vitesse exécution et de la pérennité du langage C.

Tableau 1 - API RDF/OWL en C

	Redland 1.0	3Store 2.2	Tap 0.75	RDFStore
Dernière version	02/2005	11/2004	04/2003	08/2004
Nombre de développeurs	1	5	8	3
Nombre de téléchargements (jan. 2005)	-	420	-	-
Mailing-list	Oui	Oui	Oui	-
Licence	GPL, LGPL, Apache	GPL	BSD	BSD
Paradigme de requête	Triplet	Modèle RDQL, OKBC	Triplet, Resource GetData	Modèle RDQL, SPARQL
Stockage des modèles	Mémoire, BD, Fichier	BD	Fichier	Mémoire, BD, Fichier
Bases de données supportées	Berkeley, MySQL, SQLite	MySQL		Berkeley
Formats de sérialisation	RDF/XML, N-triples, Turtle, RSS		RDF/XML	RDF/XML, N-triples, XML (DAWG-XML)
Moteur d'inférence	-	-	-	RDFS
Serveur RDFS	-	Oui	Oui	Oui
Autres fonctionnalités	Liaisons C#, Java, Obj-C, PHP, ...	-	Client Java et Perl	Liaison Perl

<sup>34</sup> <http://www.wiwiw.fu-berlin.de/suhl/bizer/toolkits/#proprietary>

Après les API pour le langage C, voici les quelques **API Java**. Java est a présent un langage incontournable : il atteint les performances du C avec la portabilité en plus. En effet, la machine virtuelle de Java peut fonctionner sur n'importe quelle système d'exploitation et sur n'importe quelle plateforme matérielle.

Le cycle de développement des applications JAVA et l'utilisation d'une grande variété de plateformes ne requiert alors aucune recompilation ni adaptation.

Voici les Frameworks RDF/OWL sous Java les plus utilisés :

Tableau 2 - API RDF/OWL pour Java (1/2)

	HP Jena 2.3	Sesame 1.2	Kaon 1.2	Kaon 2
Dernière version	10/2005	05/2005	04/2003	10/2005
Nombre de développeurs	11	12	12	-
Nombre de téléchargements (jan. 2005)	24600	6100	14200	-
Mailing-list	Oui	Oui	-	-
Licence	BSD	LGPL	LBSD	Commerciale
Paradigme	Triplet, Resource, Ontologie	Triplet, Resource	Triplet, Resource, Ontologie	Ontologie
Langage de requêtage	RDQL, SPARQL	RDQL, RQL, SeRQL	KaonQuery	-
Stockage des modèles	Mémoire, BD, Fichier	Mémoire, BD, Fichier	Mémoire, BD	Mémoire
Bases de données supportées	MySQL, PostgreSQL, Oracle, Berkeley, SQL Server, DB2	MySQL, PostgreSQL, Oracle, SQL Server	Bases compatibles SQL2	-
Formats de sérialisation	RDF/XML, N-triples, N3, Turtle	RDF/XML, N-triples, N3, Turtle	RDF/XML	RDF/XML, OWL XML
Moteur d'inférence	RDFS, OWL-Lite, DIG 1.1	RDFS, OWL-Lite	RDFS, OWL-Lite	RDFS, OWL-DL (en partie), SWRL
Serveur RDFS	Oui (Joseki)	-	Oui	Oui
Autres fonctionnalités	-	-	Extensions	

Nous pouvons constater que ces API Java bénéficient d'un nombre important de développeurs et d'utilisateurs. Deux Frameworks se démarque du lot :

Jena et Kaon sont les API les plus utilisées dans le développement d'applications utilisant le Web Sémantique ; ces dernières proposent des mises à jour régulières et des supports intéressants (Forums, mailing-list, ...). La première est considérée comme la « référence » des API RDF/OWL ; Ceci, probablement du fait qu'elle est OpenSource et qu'elle est développée au sein des laboratoires du géant Hewlett-Packard.

Plus considérés comme « Outsider », voici quelques API moins utilisées :

Tableau 3 - API RDF/OWL pour Java (2/2)

	Euler	OWLJessKB	IBM IODT	Protégé OWL API
Dernière version	05/2005	01/2005	06/2005	06/2005
Nombre de développeurs	4	1	1	1
Nombre de téléchargements (jan. 2005)	1050	-	-	-
Mailing-list	-	Oui	-	Oui
Licence	W3C	GPL	-	Mozilla
Paradigme	Modèle	Ontologie	Modèle	Ontologie
Langage de requêtage	N3QL	Jess	SPARQL	-
Stockage des modèles	Mémoire	Mémoire	Mémoire, BD	Mémoire, BD
Bases de données supportées	-	-	DB2, Derby	-
Formats de sérialisation	N3	RDF/XML	RDF/XML, N-triples, N3, TriX, TriG	RDF/XML
Moteur d'inférence	RDFS, OWL-Lite	RDFS, OWL-Lite	RDFS/OWL	RDFS/OWL
Serveur RDFS	Oui (Joseki)	-	Oui	Oui
Autres fonctionnalités	-	-	-	-

Pour des environnements de type PHP ou Prolog, il existe également des API que nous ne décrirons pas ici mais qui sont tout aussi riches et proposent des fonctionnalités intéressantes :

- RAP 0.9 qui est un outil disponible pour PHP. Celui-ci est régulièrement mis à jour par une équipe de 7 développeurs. C'est « l'outil » officiel RDF/OWL des développeurs PHP.
- SWI-Prolog Semantic Web Lib 5.4, extension de Prolog, qui est le langage de programmation logique basé sur des axiomes. Il gère donc parfaitement le RDF.

## (2) Jena<sup>35</sup>



Comme nous le disions précédemment, Jena est le Framework de « référence » RDF/OWL. Sa notoriété n'est pas seulement due au fait qu'il est développé par le géant californien; Jena offre également toutes les fonctionnalités qu'attendent le développeur et ceci avec flexibilité et d'excellentes documentations.

Fruit du programme HP Labs Semantic Web, Jena reste un projet OpenSource ; il permet de manipuler du RDF, RDFS et OWL selon trois formats (XML, N3 ou N-triples), accède au modèle en mémoire ou dans une base de données relationnelles, inclus un moteur d'inférence simple (moteur à chaînage avant, arrière ou mixte) et utilise un langage de requête : le RDQL.

Outre ces nombreuses fonctionnalités de base, Jena peut utiliser l'interface DIG : ceci permet d'éviter l'utilisation du micro-raisonneur intégré et permet alors d'exploiter un raisonneur OWL dédié de type Racer ou Pellet.

► Jena est le framework de référence pour le Web Sémantique, il propose toutes les fonctionnalités nécessaires à l'élaboration d'applications basé sur le Web Sémantique.

Pour la publication de documents RDF sur la toile, l'outil Joseki<sup>36</sup> est le compagnon parfait de Jena. Il s'intègre à un serveur d'application type Tomcat et permet aisément le partage d'ontologies.

---

<sup>35</sup> <http://jena.sourceforge.net>

<sup>36</sup> <http://www.joseki.org>



## b) Les éditeurs et outils de visualisation

On pourrait dire que les systèmes experts sont les ancêtres du Web Sémantique. Un très gros inconvénient de ces derniers vient du fait qu'il arrivait que l'on obtenait aucune explication sur les raisonnements : ces systèmes étaient des « boîtes noires ». Afin de remédier à ce problème, divers outils existent pour éditer et visualiser les connaissances structurées autour du Web Sémantique.

### (1) Protégé<sup>37</sup>



Protégé est un éditeur d'ontologies et un framework de gestion de connaissances gratuit et open source. Il est basé sur Java et est extrêmement extensible. Il peut alors servir de socle pour le développement d'applications basées sur la gestion de la connaissance.

Protégé supporte le XML Schema, RDF et RDFS ainsi que OWL. L'interface est customisable et permet modélisation et développement rapide d'applications. C'est la Stanford Medical Informatics (SMI)<sup>38</sup> qui a initié son développement en 1987, pour des besoins principalement médicaux (ex : Guide pour hypertension)

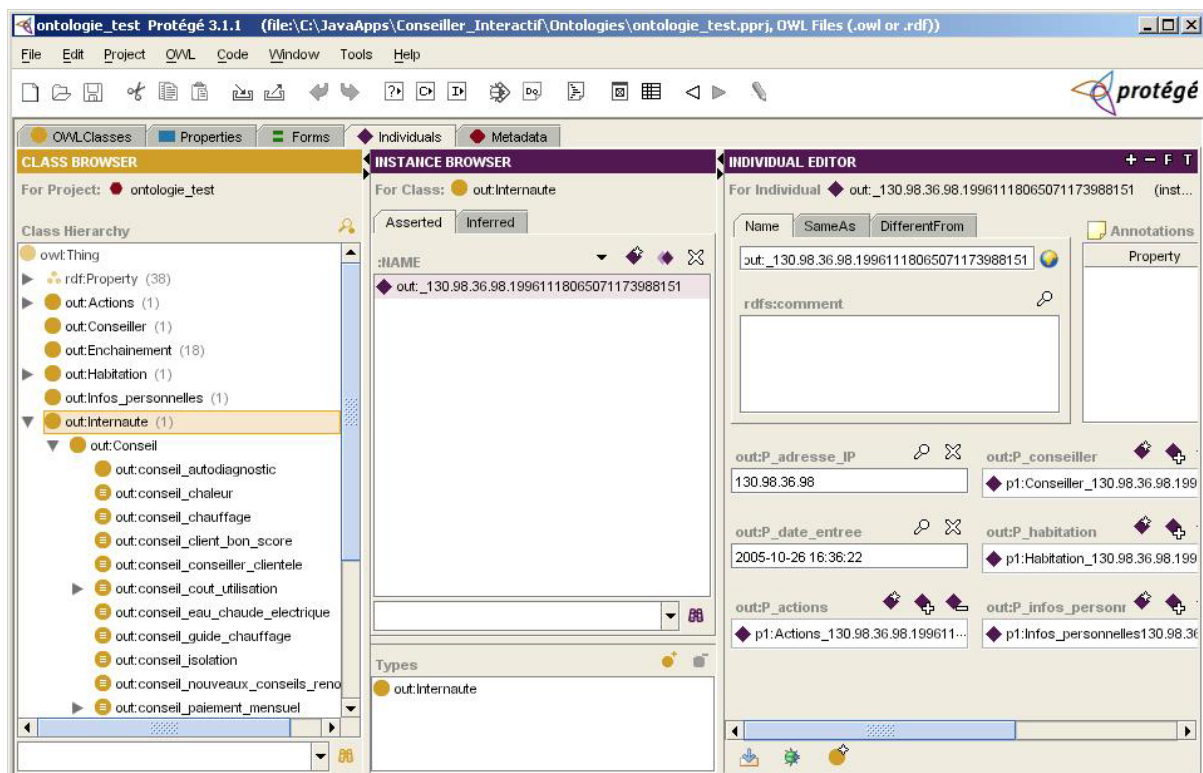


Figure 13 - Protégé

<sup>37</sup> <http://protege.stanford.edu/>

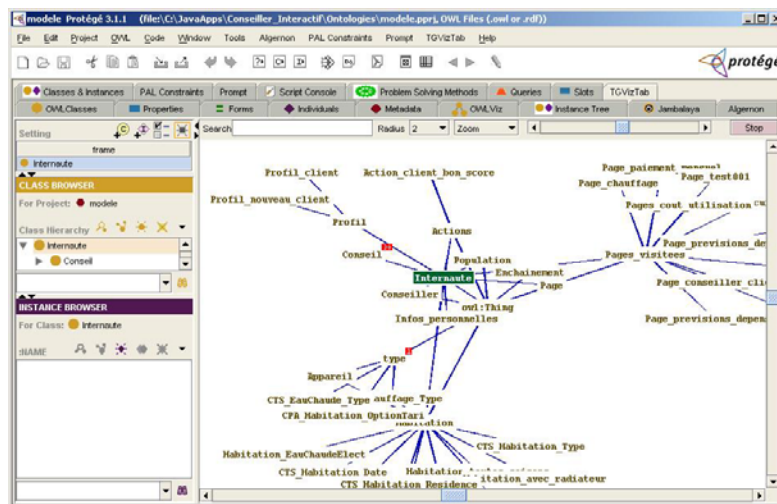
<sup>38</sup> <http://smi.stanford.edu/>

Les utilisations de Protégé sortent depuis plusieurs années de l'univers médical (ou militaire) et il est à présent l'éditeur d'ontologies par référence. Pour manipuler les ontologies, on peut utiliser l'API Protégé qui à l'instar de Jena permet des traitements de bas niveaux sur du RDF ou du OWL.

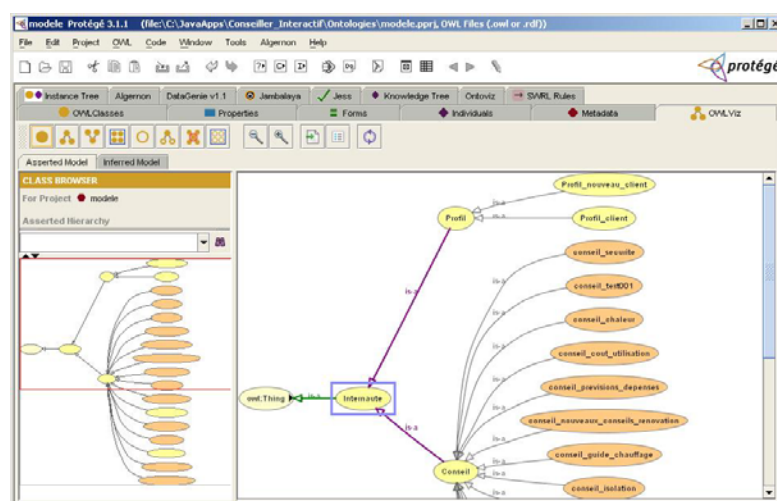
## (2) Les plugins<sup>39</sup> Protégé

Un large éventail de plugins existe pour Protégé, en voici quelques-uns :

- DataGenie qui permet la connexion à des modèles stockés sur une base de données ODBC,
- TGViz qui permet la visualisation des classes RDF sous forme d'arbres

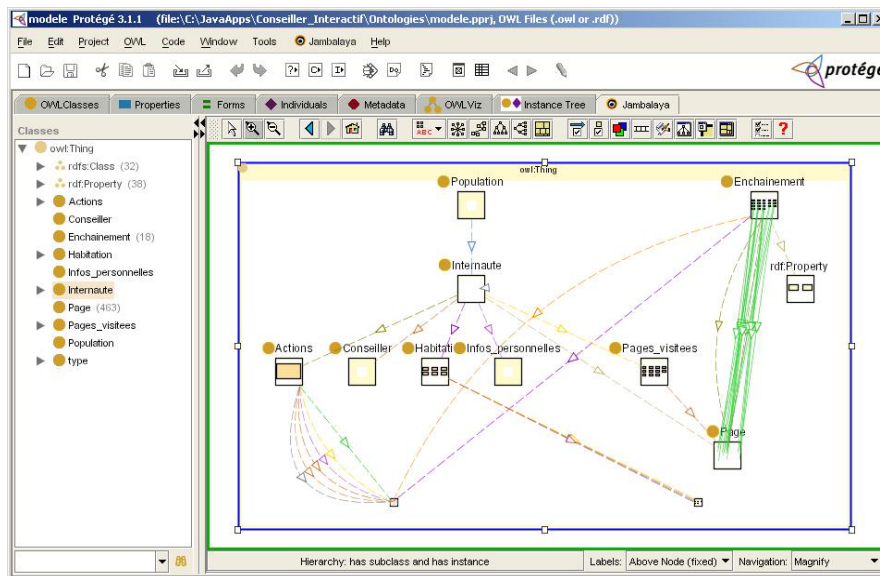


- OWLViz : Visualisation des concepts OWL



<sup>39</sup> <http://protege.cim3.net/cgi-bin/wiki.pl>

- Jambalaya : outil de visualisation très intéressant et très agréable d'utilisation, il propose une visualisation dynamique (avec des animations) et propose de fonctions de zoom très évoluer pour une vue générale ou détaillée



### (3) OilEd<sup>40</sup>

OilEd est un éditeur d'ontologies permettant à l'utilisateur la création d'ontologies DAML+OIL. Le premier objectif de OilEd était de proposer un éditeur simple qui permettait l'utilisation du langage OIL.

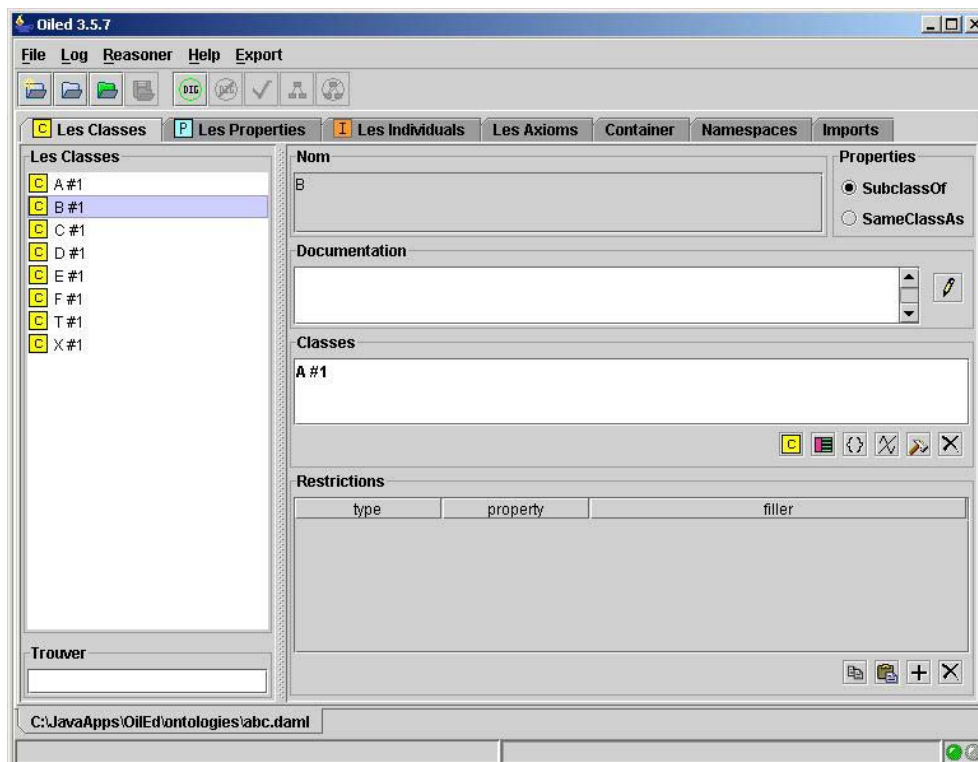


Figure 14 - OilEd

<sup>40</sup> <http://oiled.man.ac.uk/>

La version courante de OilEd ne permet pas encore de fournir un outil complet de développement d'ontologies : il ne supporte que les ontologies de taille moyenne et ne permet pas encore la migration ou l'intégration d'ontologies extérieures. Malgré cela, OilEd pense être le « Notepad » des éditeurs d'ontologies : il est en effet portable car écrit en Java et léger. Il propose néanmoins deux fonctionnalités intéressantes : la validation (« check consistency ») et l'utilisation d'un raisonneur externe (FaCT en l'occurrence).

#### (4) *Altova Semantic Works*<sup>41</sup>

Altova, éditeur de solutions XML, est surtout connu pour XMLSpy, outil industriel de référence pour tout ce qui touche à XML (modélisation, édition, débogage, transformations XSL, DTD, XSD, ...). Il présente maintenant son outil d'édition et de visualisation RDF/OWL. Semantic Works permet l'édition XML ou N-triplets des documents RDF et OWL ; la visualisation des ontologies OWL ressemble beaucoup à ce que fait XMLSpy pour les XSD, et ce n'est pas pour nous déplaire : l'écriture des ontologies est très facilitée et un module nous permet de déterminer les erreurs et le niveau du langage OWL (Lite, DL ou Full).

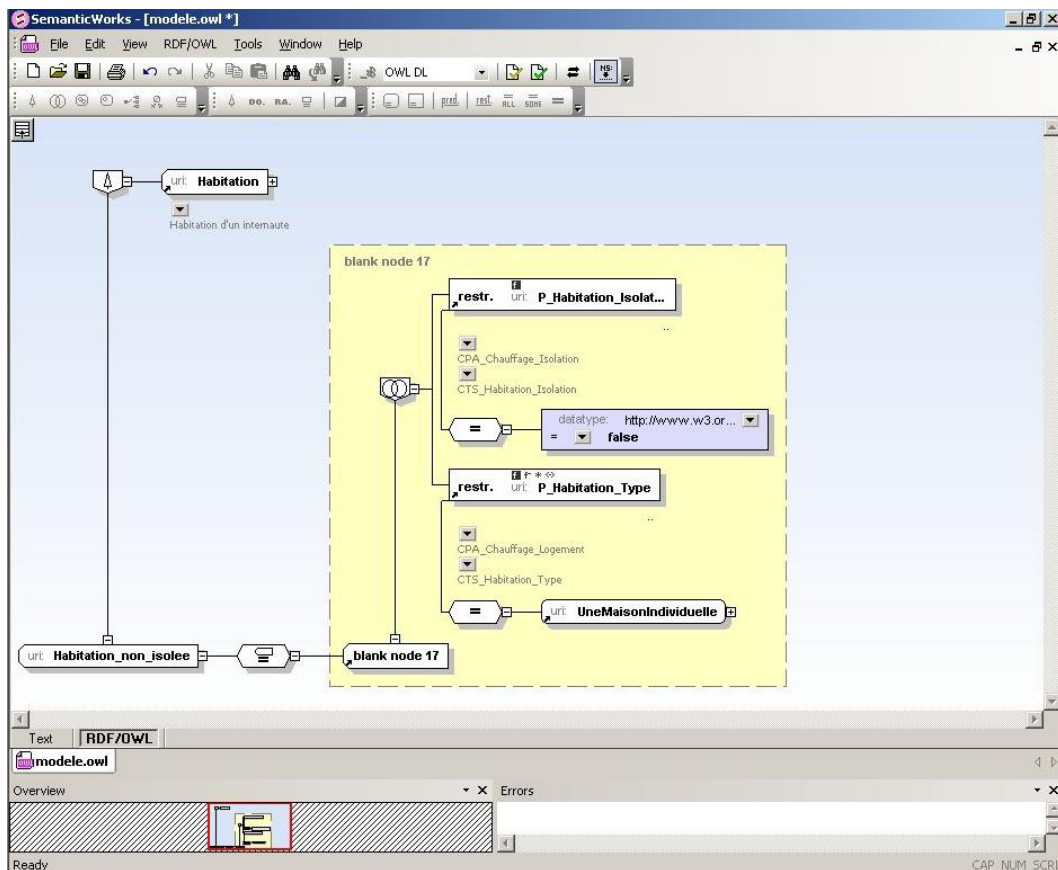


Figure 15 - Semantic Works

Au niveau fonctionnel, l'outil est très loin de Protégé car il ne propose pas de fonctions de raisonnement ni de plugins de visualisations. Finalement, ce logiciel n'est libre ni gratuit, il faudra acheter une licence pour pouvoir l'utiliser.

<sup>41</sup> [http://www.altova.com/products\\_semanticworks.html](http://www.altova.com/products_semanticworks.html)

## 7. Les raisonneurs

Nous avons utilisé deux type de raisonneurs : les moteurs à base de règles pour une implémentation rapide du premier prototype et les raisonneurs OWL pour les prototypes suivants.

### a) Les moteurs à base de règles

Il existe trois types de moteurs à base de règles :

1. Les moteurs de type « Forward chaining » ou chaînage avant qui sont des moteurs utilisant la plupart du temps l'algorithme de RETE<sup>42</sup>.

Ces moteurs utilisent les faits et les règles pour dériver de nouveaux faits qui seront ajoutés à la base. On déclenche alors toutes les règles dont les prémisses sont vraies et.

2. Les moteurs de type « Backward chaining » ou chaînage arrière.  
Le principe consiste à réaliser (vérifier) un but donné. Le moteur sélectionne alors un ensemble de règles adéquates pour vérifier le « but » et ensuite déclenche une règle choisie parmi cet ensemble. Cette dernière définit des sous-buts qui seront vérifiés par le système.
3. Les moteurs hybrides ou par chaînage mixte utilisent les deux stratégies précédentes. Cette stratégie permet de réduire l'arbre de déduction.

Dans un premier lieu, le prototype utilisait le moteur à chaînage avant de Jena :

Exemple de règles:

```
[r2_chauffage:
  (?internaute s:P_pages_visitees ?listePages)
  (?internaute s:P_actions ?actions)

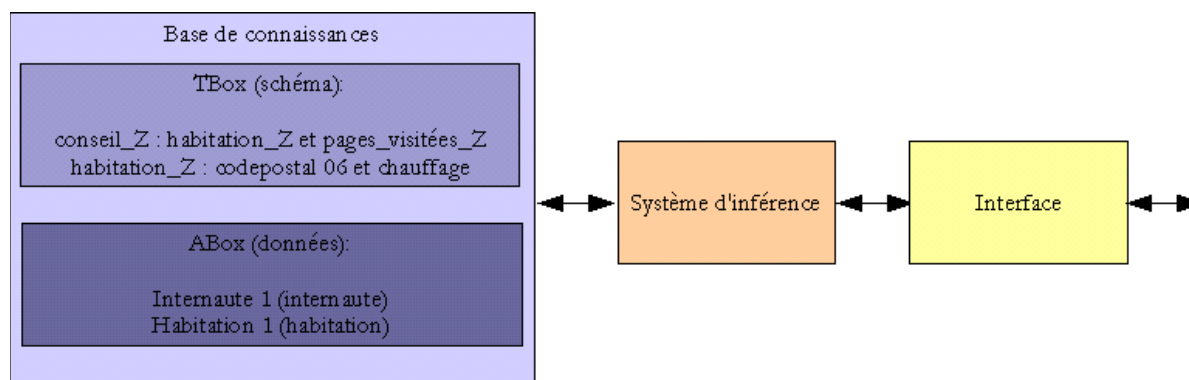
  (?listePages ?li_0 'rubrique166.html')
  (?listePages ?li_1 'rubrique23.html')

  -> (?actions s:P_a_executer 'action_chauffage_cas2')
```

<sup>42</sup> C.L. Forgy, RETE: A fast algorithm for the many pattern/many object pattern match problem, Artificial Intelligence 1982

## b) Les raisonneurs OWL

Voici un schéma présentant la mécanique interne des raisonneurs OWL.



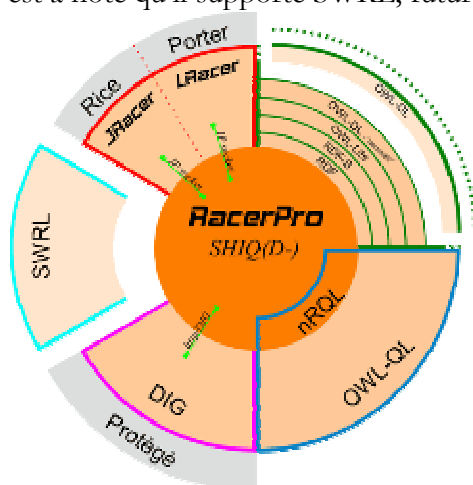
La base de connaissance est constituée de deux blocs : le bloc TBox contenant les règles et le bloc ABox contenant les données sur lesquelles on va raisonner, cette base est connectée au système d'inférence (implémenté la plupart du temps avec des moteurs de règles), relié à une interface standard (ex : protocole DIG) ou non (propriétaire) cachant le cœur du système.

### (1) *Racer*<sup>43</sup>

RACER ou Renamed ABox and Concept Expression Reasoner, premier raisonneur, propose deux types d'interface : une interface Java (JRacer) et l'interface DIG. Ce raisonneur est capable d'effectuer des raisonnements pour les trois langages OWL et était gratuit jusqu'à la version 1.7.23. Un grand avantage de Racer est sa possibilité d'utilisation en cluster. Les premiers tests de raisonnements OWL du prototype utilisaient cette version de Racer avec une connexion DIG.

### (2) *RacerPro*<sup>44</sup>

RacerPro est la nouvelle génération de raisonneurs Racer. Depuis la version 1.8, il devient payant (une licence coûtant 10 000€). Il propose beaucoup plus de fonctionnalités en terme d'interfaçage (JRacer pour du Java, LRacer pour du Lisp, DIG) et des outils de visualisation (RacerPorter) et d'optimisation (RacerPlus). Il est à noter qu'il supporte SWRL, futur langage de règles de la W3C.



<sup>43</sup> [www.cs.concordia.ca/~haarslev/racer](http://www.cs.concordia.ca/~haarslev/racer)

<sup>44</sup> <http://www.racer-systems.com/>

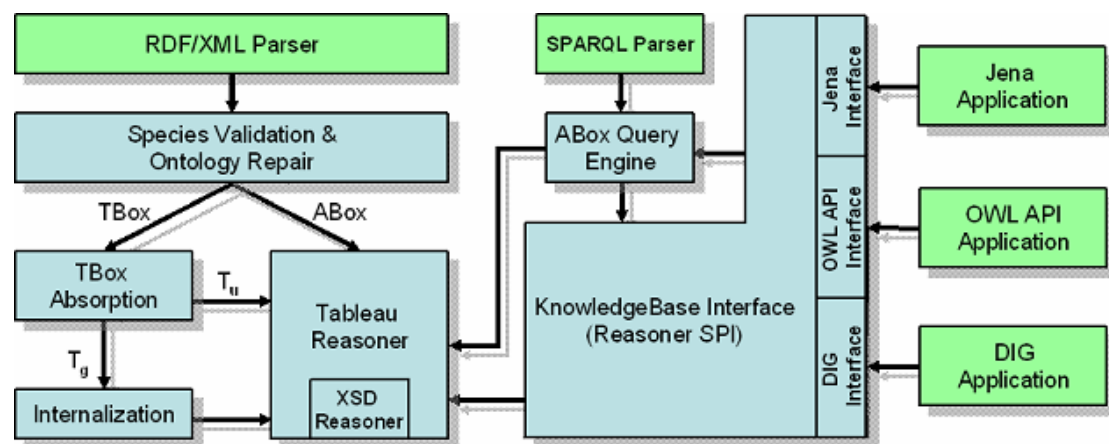
### (3) Pellet<sup>45</sup>

Pellet est raisonneur OWL DL OpenSource. Il est utilisable avec Jena et OWL API et fournit également une interface DIG.

Pellet offre des fonctionnalités intéressantes :

- L'analyse et la correction des langages OWL DL et OWL Full,
- La validation de schéma,
- Le support des XML Schéma dans les ontologies (entiers, string, date ou types définis par l'utilisateur),
- Le raisonnement sur plusieurs ontologies (E-connections)
- Le débogage d'ontologies et la détection des concepts impossible à satisfaire (concept pouvant amener à un plantage du raisonneur quand il n'est pas détecté).

Voici l'architecture interne de Pellet :



Pellet est le raisonneur par défaut du prototype. En effet, la dernière version de Pellet (la version 1.3) est très performante et dépasse très largement les autres raisonneurs.

### (4) FaCT<sup>46</sup> et FaCT++<sup>47</sup>

FaCT et FaCT++ sont des raisonneurs OWL DL développés respectivement en Java (servlet) et en C++ par l'Université de Manchester. Les versions actuelles sont compatibles DIG, fonctionnent avec les API Protégé mais pas entièrement avec l'API Jena. La version 1.0 de FaCT++ qui devrait disponible dans les prochains jours et devrait alors corriger ce problème.

<sup>45</sup> <http://www.mindswap.org/2003/pellet/>

<sup>46</sup> <http://www.cs.man.ac.uk/~horrocks/FaCT/>

<sup>47</sup> <http://owl.man.ac.uk/factplusplus/>

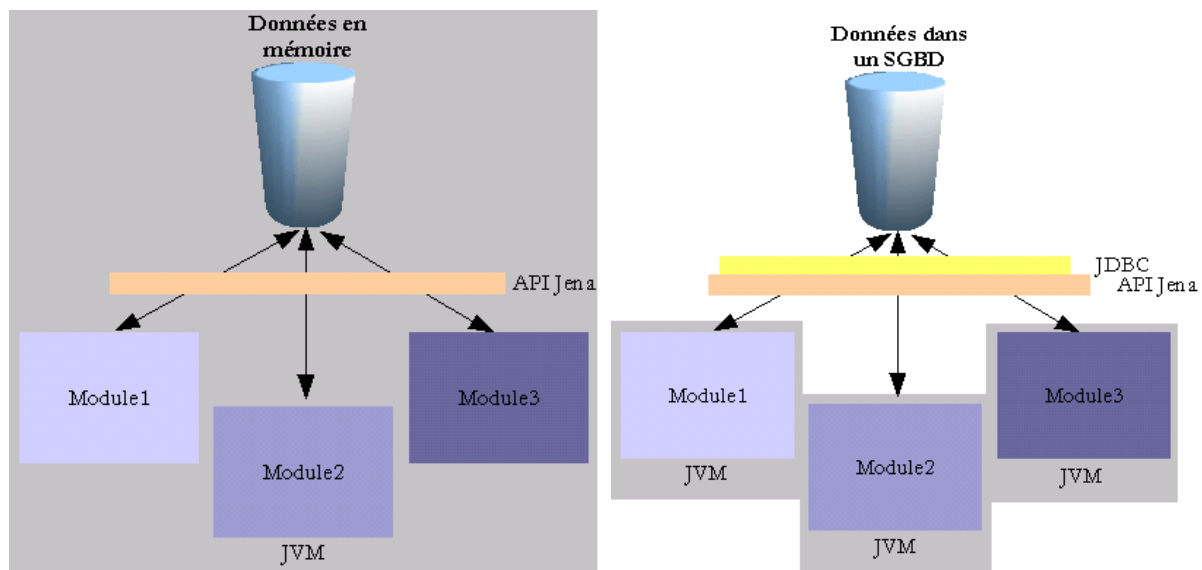
## 8. Les protocoles de communications

Nous allons présenter ici les diverses méthodes de communication entre les différents modules : ces communications peuvent être effectuées entre deux composants d'un même module (par exemple le raisonneur qui utilise DIG) ou entre la machine de l'internaute et le serveur d'application (AJAX).

### a) Jena : un intergiciel

Jena n'est pas, à proprement parlé, un intergiciel car sa principale raison d'être est la manipulation de documents RDF et OWL. Cependant, Jena propose des fonctionnalités permettant à notre application d'être répartie : si on utilise un stockage en base de données et un raisonneur connecté en utilisant DIG, il suffit de quelques modifications à notre code pour qu'il puisse fonctionner indifféremment sur une machine ou plusieurs machines.

L'API Jena nous cache alors toutes les communications utilisant HTTP et la couche JDBC.



### b) DIG

DIG (DL Implementation Group) est le protocole d'interconnexion entre les API OWL et les raisonneurs OWL. Il est basé sur HTTP et XML de la même manière que SOAP ou XML-RPC.

Il y a deux types de messages DIG :

- Les messages « Request » envoyés par le client en utilisant une requête HTTP Post :
  - Le message « identification »,
  - Le message « management »,
  - Le message « tells » : qui donne au raisonneur les concepts sur lequel il faudra raisonner,



- Le message « ask » : qui demande au raisonneur la résolution d'une question.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<tells
xmlns="http://dl.kr.org/dig/2003/02/lang"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://dl.kr.org/dig/2003/02/lang
http://dl-web.man.ac.uk/dig/2003/02/dig.xsd"
uri="urn:uuid:abcdefgh-1234-1234-12345689ab">
  <defconcept name="driver"/>
  <equalc>
    <catom name="driver"/>
    <and>
      <catom name="person"/>
      <some>
        <ratom name="drives"/>
        <catom name="vehicle"/>
      </some>
    </and>
  </equalc>
  <defconcept name="person"/>
  <defconcept name="vehicle"/>
  <defrole name="drives"/>
</tells>
```

- Les messages « Response » envoyés du serveur vers le client suite à un message « Request ».

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<response
xmlns="http://dl.kr.org/dig/2003/02/lang"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://dl.kr.org/dig/2003/02/lang
http://dl-web.man.ac.uk/dig/2003/02/dig.xsd">
  <ok>
    <warning message="Undeclared Name"
code="99">
      Class Cat used but not declared
    </warning>
  </ok>
</response>
```

### c) Ajax

Ajax ou Asynchronous JavaScript And XML est une technologie qui existe déjà de plusieurs années, elle permet d'effectuer des requêtes HTTP et faire des traitements de type XML dans les pages Web en tâche de fond du navigateur.

AJAX est souvent utilisé pour éviter le rafraîchissement d'une page : quand on recharge une page, le contenu est effacé puis rechargé.

Il existe maintenant des API permettant l'utilisation d'Ajax avec des environnements de développement Web type J2EE, ce qui simplifie considérablement le travail à faire.

```
function getXMLObject(){
  if(window.XMLHttpRequest) {
    objXml = new XMLHttpRequest();// Firefox
  } else if(window.ActiveXObject) {
    objXml = new ActiveXObject("Microsoft.XMLHTTP");// IE
  }
  return objXml;
}

function lireFichier(){
  //contenu_fichier
  objXml = getXMLObject();

  objXml.open("GET", url, true);
  objXml.onreadystatechange=function() {
    if (objXml.readyState==4) {
      chargePage(objXml.responseText);
    }
  }

  objXml.send(null);
}
```

#### d) Les Web Services



Les services Web fournissent l'interopérabilité entre logiciels et plateformes hétérogènes. Les standards et protocoles utilisés dans les services Web permettront aux applications EDF d'interagir avec la plateforme du « Conseiller Interactif ».

On pourra en conséquence échanger des informations concernant un client avec SIMM. Ceci afin d'enrichir son profil et ainsi lui proposer des conseils encore plus pertinents.

Le « Conseiller Interactif » disposera d'une interface Web Services basé sur Axis<sup>48</sup> de la fondation Apache.

---

<sup>48</sup> <http://ws.apache.org/axis/>

## V. Le Conseiller Interactif

Je vais vous présenter ici ma contribution à l'action « Conseiller Interactif ». J'ai eu à faire des choix techniques, des choix d'architectures, de la modélisation et beaucoup de développement.

### *A. Cahier des charges*

A partir de la note de cadrage et de plusieurs réunions avec Stéphane Rouilly, Pascale Capron et Pierre Chaussecourte, il a été possible de définir un cahier des charges définissant les fonctionnalités auxquelles doivent répondre le Conseiller Interactif :

- **Intégration au Serveur Web Particuliers** pour un recueil d'informations concernant l'internaute :
  - Des informations concernant ses pages visitées,
  - Des informations déclaratives provenant des formulaires.
- **Remise des conseils** à l'internaute en fonction des informations que l'on a de lui et en fonction d'une **base de conseils**,
- **Capitalisation** des informations recueillies et inférées,
- Utilisation du **Web Sémantique** comme cœur du système de raisonnement,
- **Modularité** de la plateforme : interchangeabilité des composants,
- **Explicitation** du raisonnement,

### *B. La plateforme du Conseiller Interactif*

Le principal objectif du Conseiller Interactif est de fournir, en temps réel, des conseils en fonction des informations que nous avons sur le client. Les diverses informations nous sont données par l'internaute et par son « surf » et par une connexion à un système d'informations.

Lors du recueil des charges, il a été identifié divers besoins dont celui de faire intervenir des experts de chaque domaine (marketing, Webmastering, architecte en S.I.) pour l'élaboration du cœur du Conseiller Interactif. Chaque spécialiste doit alors amener son savoir et le capitaliser dans un système d'information spécialisé, mon travail était alors de fournir des structures de données interconnectées ayant chacun son cycle de vie.

### C. Architecture métier

Voici un schéma représentant les divers acteurs et les services que doivent proposer la plateforme du conseiller interactif. L'idée principale de l'architecture est la séparation « physique » des acteurs (en rouge) et le partage des référentiels métiers (en bleu) :

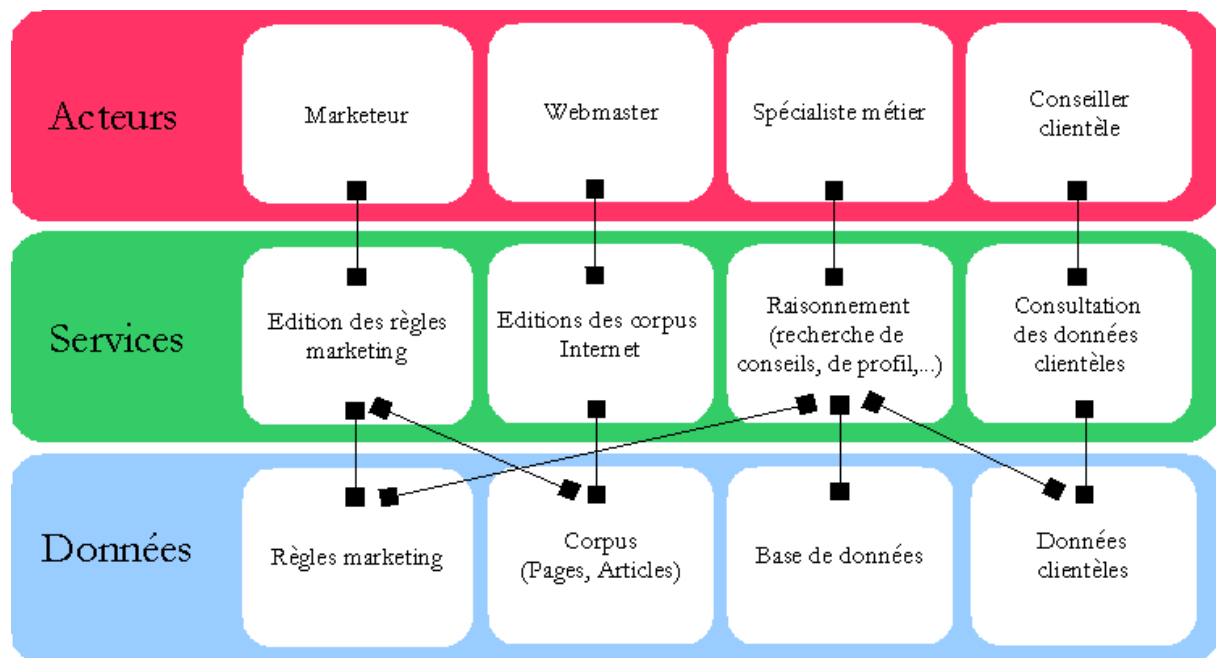


Figure 16 - La séparation des acteurs et des référentiels

La plateforme du conseiller interactif fait intervenir quatre types d'experts :

1. Le *marketeur* ayant une forte connaissance sur le métier de l'entreprise et sur la clientèle (sur ses besoins), cette personne va rédiger les règles marketing.
2. Le *webmaster* ou responsable du site Internet chargé de mettre à jour les corpus contenus sur le site : ex : Nouvelle page à référencer.
3. Le *spécialiste métier* qui peut consulter le fonctionnement du « conseiller interactif » : les conseils prodigués et visualiser la base de données clientèle construite.
4. Un *conseiller clientèle* utilisant un système d'information pour obtenir un supplément d'informations sur un client.

Ces quatre types de spécialistes vont alors assurer des modifications et évolutions sur chaque composant métier sans avoir à s'occuper des autres parties de la plateforme. (ex : Le Webmaster ajoute des nouvelles pages sans se préoccuper des règles marketing).

## ***D. Choix des technologies***

Afin d'obtenir une plateforme robuste, rapide et évolutive, plusieurs choix ont été faits sur les aspects technologiques du Conseiller. Nous allons vous présenter ces sélections.

### **1. Architecture du Conseiller**

Le « Conseiller Interactif » est une application Web utilisant le Web Sémantique, il devra supporter des contraintes de charges importantes, se connecter au serveur Web Apache et utiliser BEA Weblogic (référentiel EDF). Nous utiliserons donc l'environnement Java ainsi qu'un serveur J2EE pour l'ensemble des développements.

### **2. Serveur d'application**

Il ne m'a été imposé aucune technologie pour le Conseiller Interactif. Néanmoins, le serveur Web doit être le même que celui du serveur en exploitation : le serveur référence de la fondation Apache : Apache HTTP Server 1.3.

Ce dernier intègre en standard les modules permettant de faire du Web dynamique : Perl ou encore en PHP, mais j'ai préféré opté pour un serveur d'application de type Java : ceci afin de bénéficier de la portabilité de l'environnement et du grand nombre de « frameworks » OpenSource ou non disponibles pour cette technologie.

Le serveur d'application Java devra être un moteur de Servlet/JSP véloce et fiable; nous avons le choix entre plusieurs solutions, trois projets OpenSource sortent du lot :

- Jakarta Tomcat (<http://jakarta.apache.org>), la référence d'implémentation des spécifications Java Servlet et JavaServerPages de Sun. Il serveur HTTP, de conteneur de Servlet et est compatible J2EE ; cette dernière fonctionnalité permet alors de porter les développements sur des serveurs d'applications commerciaux du type WebSphere ou Weblogic. Tomcat est le conteneur de Servlet le plus utilisé dans le cadre d'applications Web en Java.,
- Resin (<http://caucho.com/resin/>) est à l'instar de Tomcat un serveur d'application performant tenant la charge ; il dispose de fonctionnalités intéressantes : Load balancing, Clustering, ou encore SSL intégré.
- Jetty (<http://jetty.morabay.org>), est un serveur d'application 100% Java. C'est un moteur très léger qui n'offre cependant qu'une tenue en charge très limitée. Néanmoins, le serveur J2EE OpenSource JBoss l'utilise comme conteneur par défaut.

Au vu de son excellent support de charge, de son implémentation des spécifications JSP 2.0 et Servlet 2.4 ainsi que sa facilité de portabilité, nous décidons alors de prendre comme conteneur : **Jakarta Tomcat.**

### 3. La base de données relationnelle

La base de données joue un rôle important dans notre plateforme, celle-ci va contenir toutes les informations que nous avons du « surfeur ». En terme de contraintes techniques, la seule chose nécessaire est la possibilité de connecter cette base à des services Java : il est donc nécessaire que les concepteurs de la base de données fournissent des API compatibles JDBC<sup>49</sup>.

Mon choix s'est porté sur les deux bases de données OpenSource très utilisées et connues pour leurs fiabilités : **MySQL** et **PostgreSQL**. Tout deux disposent de connecteurs JDBC, MySQL est connu pour sa simplicité d'installation et sa facilité d'utilisation par l'intermédiaire de l'interface Web nommée « PHPMyAdmin » ; Quand à PostgreSQL, elle est tout simplement la SGBDR la plus fiable du domaine libre, elle est plus adaptée que MySQL pour une utilisation industrielle.

#### E. Modélisation des connaissances

##### 1. Le modèle « Internaute »

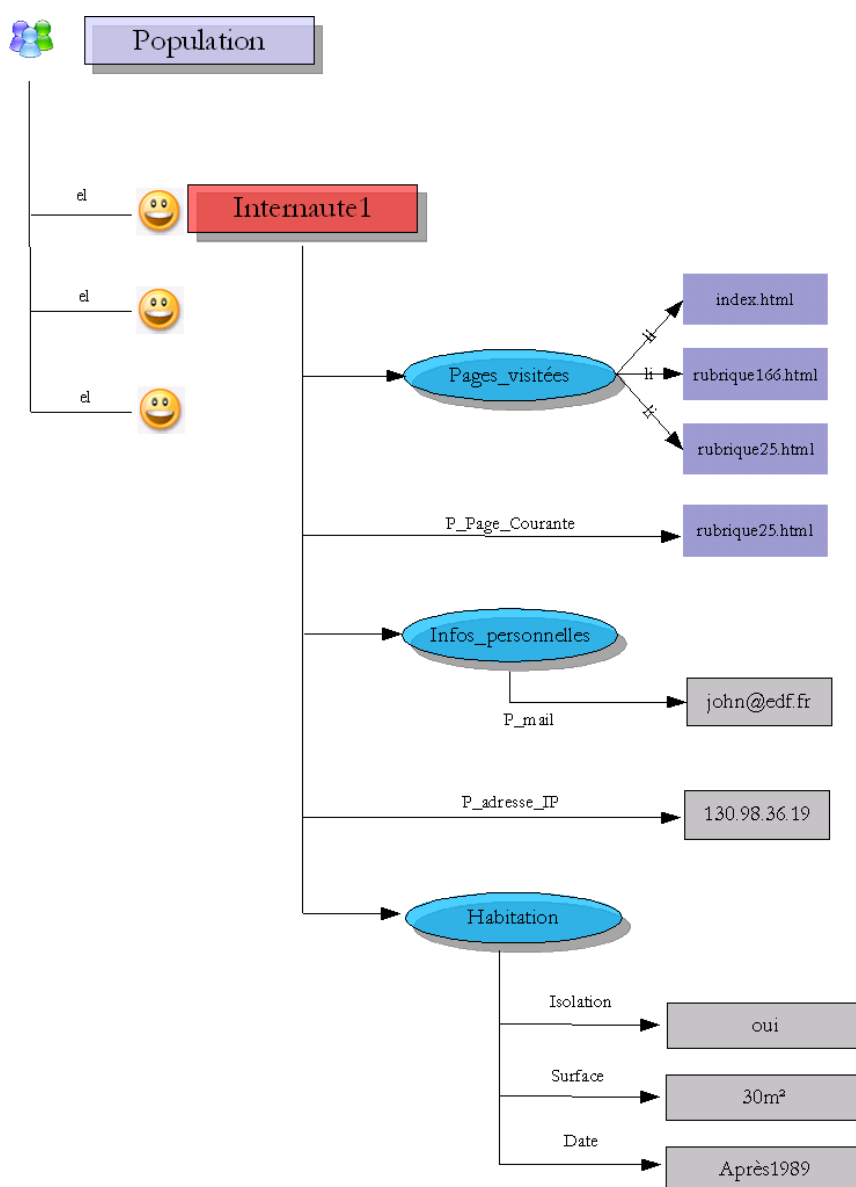
En m'aidant de la note de cadrage et de l'inventaire des services contenus dans les « services interactifs », j'ai établi un schéma RDF représentant l'internaute :

La « population » des internautes comprend (propriété « el ») des individus « Internaute ».

Un internaute :

- possède une liste de « Pages visitées », chaque page étant rattaché à ce container par la propriété « li ».
- peut posséder des informations personnelles comme un e-mail.
- possède une habitation avec une multitude d'informations : (isolation, date de construction, surface, appareil électrique, ...)

Un internaute est sur une page courante (la dernière page visitée), possède une adresse IP, et une date d'entrée sur le site internet.



<sup>49</sup> Java DataBase Connectivity

La modélisation utilise RDF Schema et OWL, nous spécifions :

- Les classes (Population, Internaute, Habitation),
- Les propriétés (el, li, Page\_Courante, P\_Isolation, ...) : précédée d'un **P\_** (pour faciliter la lecture)
- Les valeurs des propriétés ou leurs types (booléen, string, énumération,...)

La modélisation ne tient compte que des informations que l'on a du client à partir du site Internet, une extension prévoit de rajouter certaines informations contenues dans l'Agence En Ligne (propriétaire, consommation, ...).

## 2. Une ontologie sur le contenu du site Internet

Afin d'établir une ontologie du site Internet, j'ai développé une application permettant de « scanner » l'intégralité de l'arborescence du site Internet : cette application collecte les informations contenues dans chaque page : titre et méta informations (mots-clés,...).

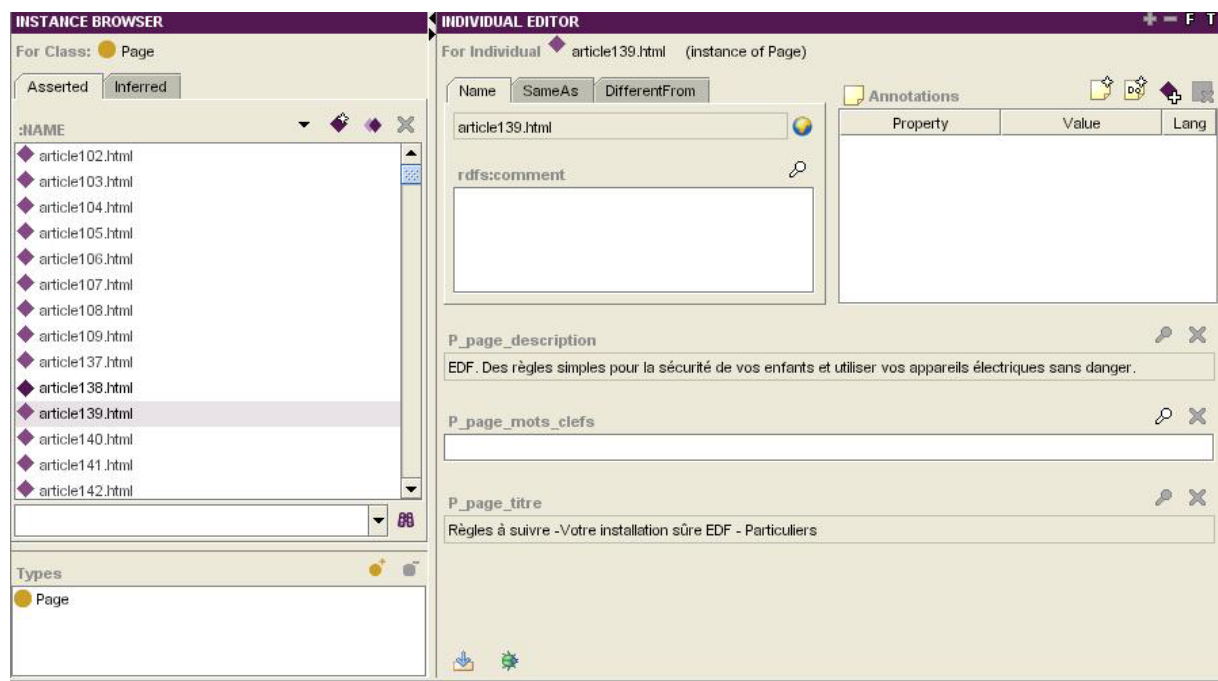


Figure 17 - Ontologie "Page"

## 3. Les règles marketing

### a) Qu'est ce que c'est ?

Les règles marketing, écrite par Pascale Capron, sont la retranscription des connaissances métiers d'un conseiller EDF dans le cœur du « Conseiller Interactif ».

Ce sont ces règles qui vont personnaliser l'information : valoriser des offres et services ou donner des suggestions comme le ferait un conseiller humain.

Voici un exemple de règle :

**SI** l'internaute est passé par la page « Vous rénovez » **ET** par la page « prêt vivrélec » **ET** a déclaré avoir le chauffage électrique **ET** que l'on sait qu'il habite dans le 06

**ALORS** afficher conseil « Pour être bien chez vous en toutes saisons, avez-vous pensé à la climatisation réversible ? »

## b) Modélisation d'un conseil

La modélisation d'un conseil utilise les modélisations précédentes : celle de l'internaute et celle des pages du site. Un « Conseil » est une sous-classe d'Internaute adjoints de contraintes.

Les contraintes sont celles de OWL-DL :

$\forall$  *allValuesFrom* : pour chaque instance de la classe, les propriétés doivent être tous membres de la classe désignée par la clause *allValuesFrom*.

$\exists$  *someValuesFrom* : pour chaque instance de la classe, au moins une des propriétés est vraie

$\in$  *hasValue* : définition d'une classe en fonction de l'existence de valeurs de propriétés particulières. L'individu appartiendra à la classe à condition qu'au moins une des valeurs de la propriété soit égale à la ressource de *hasValue*.

$=, \leq, \geq$  *cardinality, minCardinality, maxCardinality* : la cardinalité d'une propriété donné doit être égale, inférieure ou égale, supérieure ou égale à la valeur donnée.

$\cap, \cup$  *intersectionOf, unionOf* : intersection ou union de propriétés.

A partir de l'ontologie « Page », on sait que la page « Vous rénovez » a comme URI :

<http://particuliers.edf.fr/rubrique49.html> et la page « prêt vivrélec » :

<http://particuliers.edf.fr/rubrique16.html>

Il faut créer une classe « Pages\_Visitées\_conseil1 » sous classe de « Pages\_Visitées » avec les contraintes :

(li *hasValue* <http://particuliers.edf.fr/rubrique49.html>)

*union*

(li *hasValue* <http://particuliers.edf.fr/rubrique16.html>)

De la même manière, il faut crée une classe « Habitation\_conseil1 » sous classe d'« Habitation » avec les contraintes :

(P\_Habitation\_Chauffage *hasValue* « ChauffageElectrique »)

*union*

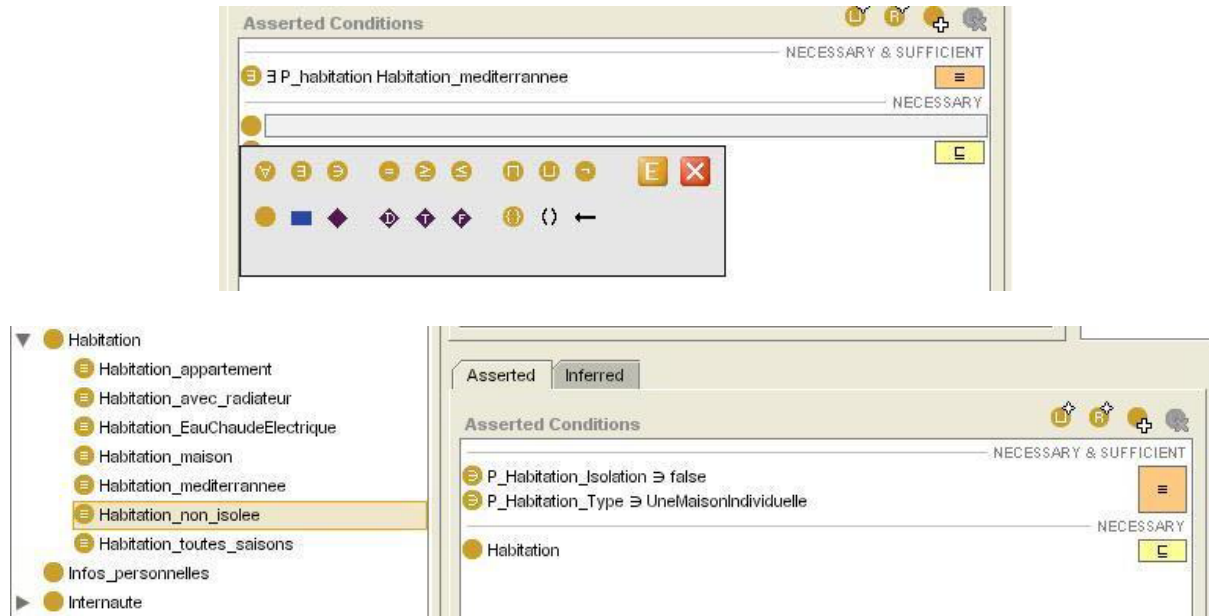
(P\_Habitation\_CodePostal *hasValue* « 06 »)

Le conseil peut maintenant être créé : « Conseil1 » est une sous classe de « Conseil » (lui-même sous classe d'Internaute) avec les contraintes :



$(\text{someValuesFrom } P\_Habitation \text{ Habitation\_conseil1})$   
*union*  
 $(\text{someValuesFrom } P\_Pages\_Visitées \text{ Pages\_Visitées\_conseil1})$

Protégé simplifie l'écriture des ses conditions :



## F. Le veilleur

Le rôle du veilleur est le recueil des informations provenant de l'internaute. En effet, durant sa navigation, l'internaute va continuellement échanger des informations avec le serveur Web : lors du chargement d'une page par l'internaute, une requête est envoyée au serveur et la page HTML est retournée par ce dernier. Il nous est alors nécessaire d'intercepter toutes les informations transitant par le serveur : le veilleur, appelé également intercepteur, est un composant logiciel chargé d'identifier les internautes et de collecter toutes les informations échangées durant la session.

### 1. Identification

Le serveur HTTP Apache peut être adjoint d'une multitude de modules et génère par défaut des fichiers logs contenant les informations sur les requêtes qui lui ont été demandées.

Premièrement, il est nécessaire d'éditer le fichier de configuration du démon Apache, **httpd.conf**, situé dans le répertoire **apache/conf/**, afin d'assurer l'identification et donc l'unicité des internautes. Pour cela, nous utiliserons le module **mod\_usertrack.so** et le configurerons afin d'assigner un Cookie, portant le nom de « Conseiller\_Interactif », à chaque internaute.

### 2. Collecte d'informations

Pour chaque requête, le serveur génère une ligne décrivant l'identité du demandeur (adresse IP), la date de la demande et son objet (la page sollicitée). Toutes ses lignes sont stockées dans des fichiers logs (**apache/logs/**).

Il est nécessaire de configurer le serveur Apache afin d'ajouter aux requêtes l'identifiant unique que nous avons précédemment créé. Lors de la génération de chaque ligne, nous rajoutons alors l'identifiant de l'internaute (stocké dans la variable `%{cookie}`) :

Le log généré se présentera alors comme suit :

**Identifiant** – Adresse IP – – *Date de connexion* – Requête – **Code réponse** – Taille de l'objet retourné

Ex :

```
130.98.36.36.52411178013754083141 130.98.36.36 - - [03/Jun/2005:14:22:55 +0200] "GET /Conseiller_Interactif/Conseiller.html HTTP/1.1" 200 644
```

### 3. Le composant de veille

Le veilleur ou intercepteur est également appelé Enrichisseur car son rôle est de reprendre les informations fournies par les fichiers log et ensuite d'enrichir notre base.

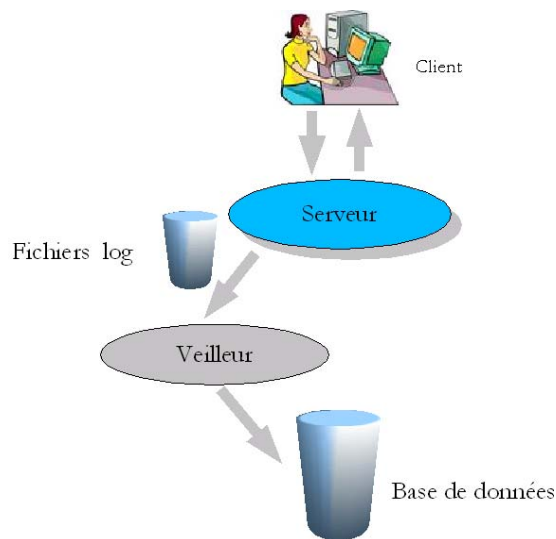


Figure 18 - Le veilleur

Le veilleur lit chaque ligne et extrait à l'aide de patterns, des expressions régulières Perl, chaque information contenue dans la requête :

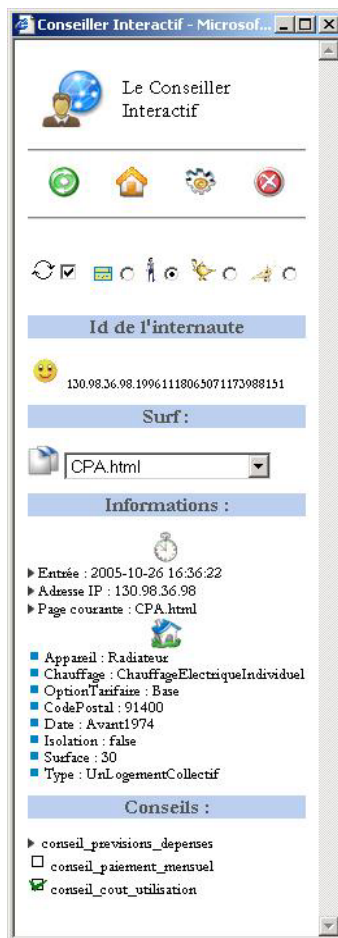
- Identifiant
- Adresse IP
- Date
- Page visitée
- Paramètres
- Nom du navigateur
- ...

Pour chaque requête valide, le veilleur ajoutera les informations relatives à l'internaute à celles déjà connues et contenues dans la base de données. Les informations sont :

- relatives au « surf » : page courante, pages visitées,
- ou des informations déclaratives envoyées depuis les formulaires du site : paramètres envoyés en HTTP GET.

### ***G. Le tableau de bord***

Le tableau de bord est l'interface nous permettant d'afficher en temps réel l'intégralité des informations recueillis par le Veilleur. Celui-ci n'est utilisé que dans le prototype du Conseiller et ne sera évidemment pas affichée dans la version industrielle.



Nous avons, de haut en bas :

1. Les options du tableau de bord :
  - a. Mise à jour
  - b. Retour au menu principal
  - c. Rechargement du Conseiller
  - d. Réinitialisation des paramètres
2. Les options du Conseiller :
  - a. Mise à jour automatique
  - b. Mode Popup
  - c. Mode Avatar – Laura
  - d. Mode Avatar – Lampe magique
  - e. Mode Avatar – Lampe fine
3. Les informations sur l'internaute
  - a. Identifiant
  - b. Pages visitées (Surf)
  - c. Information sur la connexion
    - i. Date d'entrée
    - ii. Adresse IP
    - iii. Page courante
  - d. Informations sur l'habitation
4. Conseils délivrés/en cours de délivrance (ainsi la note donnée par l'internaute)
  - a. Conseils en cours
  - b. Conseils acceptés
  - c. Conseils refusés

Figure 19 - Le tableau de bord

Le tableau de bord joue un rôle important dans l'élaboration du prototype, il nous informe en temps réel de l'enrichissement de notre profil et nous montre les conseils calculés par le raisonneur

## H. Le conseiller

Le conseiller est la face du système visible par l'utilisateur lambda. Le rôle de celui-ci est de fournir à l'internaute le conseil approprié. Conseil qui aura été précédemment « calculé » par le raisonneur et sera rajouté aux informations sur l'internaute.

Le conseiller est un ensemble de services (JSP/Servlet) stockés sur un serveur d'application Java basé sur le conteneur de Servlet. Le browser va périodiquement appeler une fonction sur le serveur d'applications pour vérifier que de nouveaux conseils sont disponibles : Ajax<sup>50</sup> est utilisé pour ce chargement en « background ».

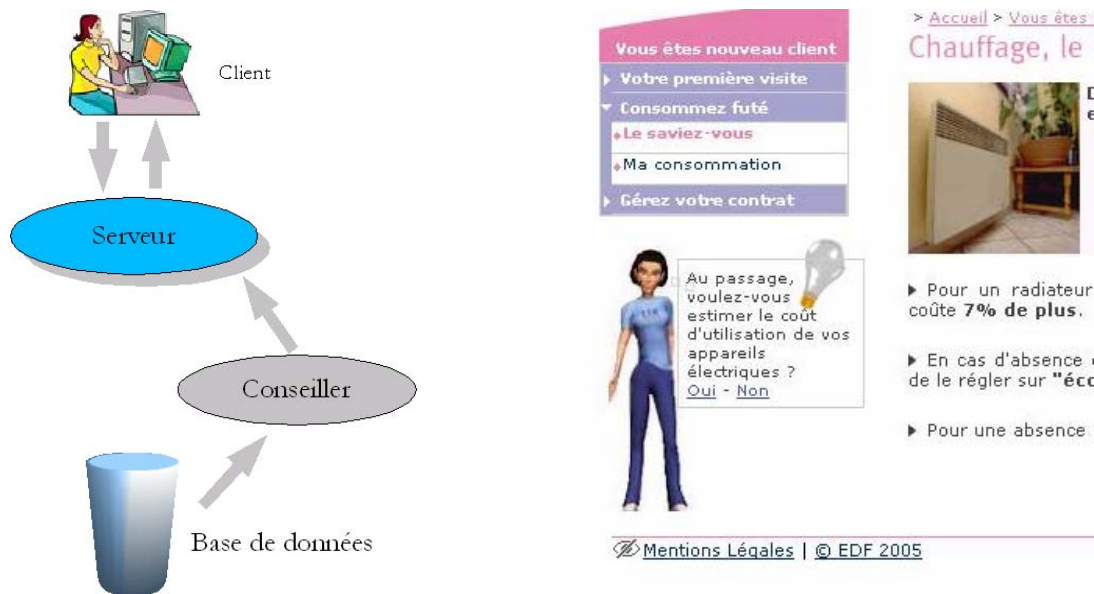


Figure 20 - Le conseiller

Le conseiller est « customisable » et peut prendre l'apparence de Laura, d'une lampe ou d'un autre type d'avatar. Il est possible de le rendre silencieux et il a été prévu dans des cas difficiles de le masquer et/ou de le transformer en bannière type publicité.

▶ Le conseiller peut se présenter de diverses manières et est entièrement intégré au site Internet.

Afin que toutes les pages Web du site particuliers affiche le conseiller, il est nécessaire de créer des scripts permettant le balayage de tous les fichiers du site et l'insertion d'un tag permettant l'affichage du « Conseiller ». L'application chargée de lire les fichiers et d'insérer le tag est appelé l'**GIntégrateur**, il est écrit en Java et utilise des expressions régulières type Perl (bibliothèque ORO<sup>51</sup>).

L'intégrateur parcourt les balises HTML de chaque fichier et insère le script à des positions différentes dans le code selon la position voulue (haute ou basse) du Conseiller dans la page Web.

<sup>50</sup> Voir Ajax

<sup>51</sup> <http://jakarta.apache.org/oro/>

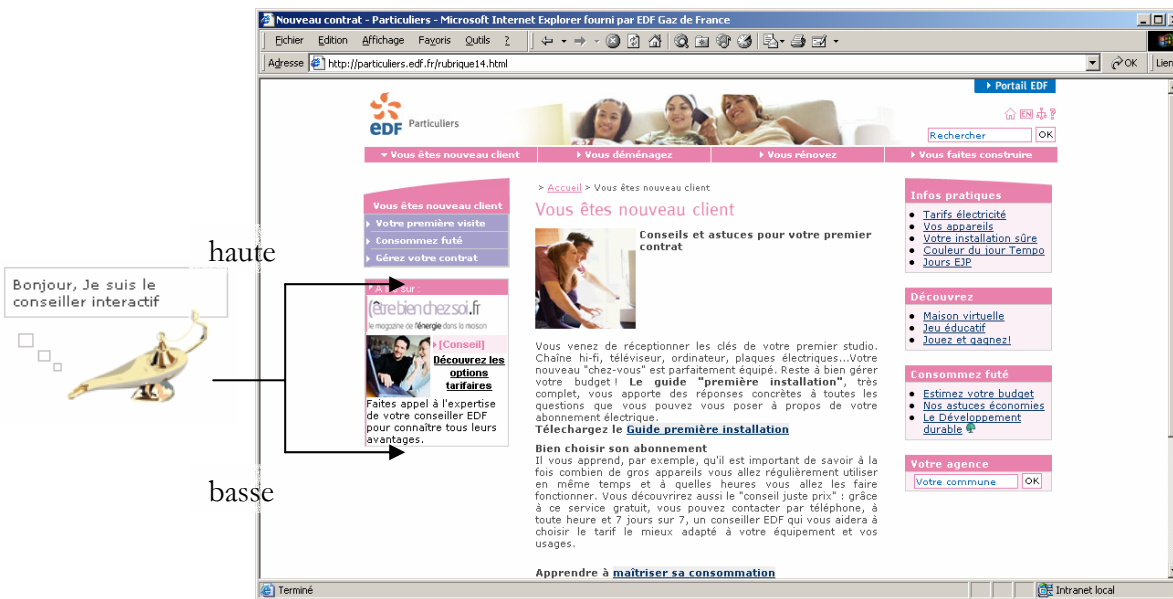


Figure 21 - Le conseiller sur une page Web

### I. Le raisonneur

Le raisonneur est au cœur de notre système. Il est chargé, à partir des informations stockées dans la base et des règles marketing, de fournir les « conseils » à l'internaute naviguant sur le site internet.

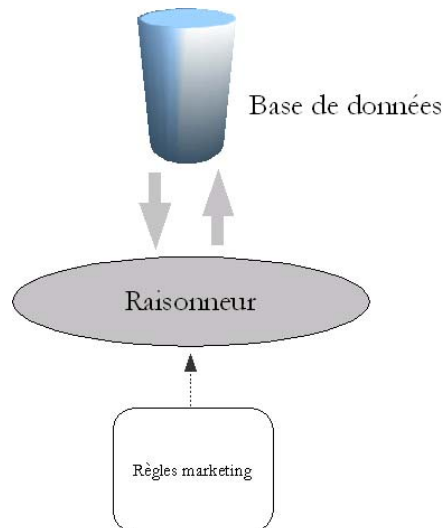


Figure 22 - Le raisonneur

Le raisonneur prend en entrée les informations contenues dans notre base de données ainsi que le modèle contenant les règles marketing. Il se connecte alors un raisonneur OWL en DIG afin d'effectuer tout les calculs d'inférences.

De nombreuses optimisations<sup>52</sup> ont été apportées au cœur du raisonneur. En effet, le raisonneur doit garantir des temps de réponse courts afin que le conseil soit donné le plus rapidement

<sup>52</sup> Voir partie Mesure des performances

possible à l'internaute. Les temps de réponse sont passés de 8 secondes par raisonnement (Juin 2005) à 700 millisecondes aujourd'hui (Décembre 2005)

## J. Les outils

Deux outils, complétant la plateforme, ont été créés dans des besoins particuliers : un outil de présentation des données et des déductions : GRaison, et un outil permettant la simulation de l'arrivée des internautes à partir de log : l'enrichisseur en orgue de Barbarie.

### 1. Visualisation des données et des déductions

Un des problèmes des Systèmes Experts était l'impossibilité de voir la mécanique interne du système : on posait la question et on avait la réponse, mais on ne savait pas pourquoi. L'idée est alors de fournir un outil qui nous permette de la même manière que le tableau de bord, d'afficher les informations sur l'internaute mais également d'y rajouter la couche « Raisonnement » et expliquer les inférences.

Cet outil présente alors les données de manière arborescente (comme le modèle RDF) et affiche à côté de chaque classe (Internaute, Habitation, Pages Visitées) les classes déduites :

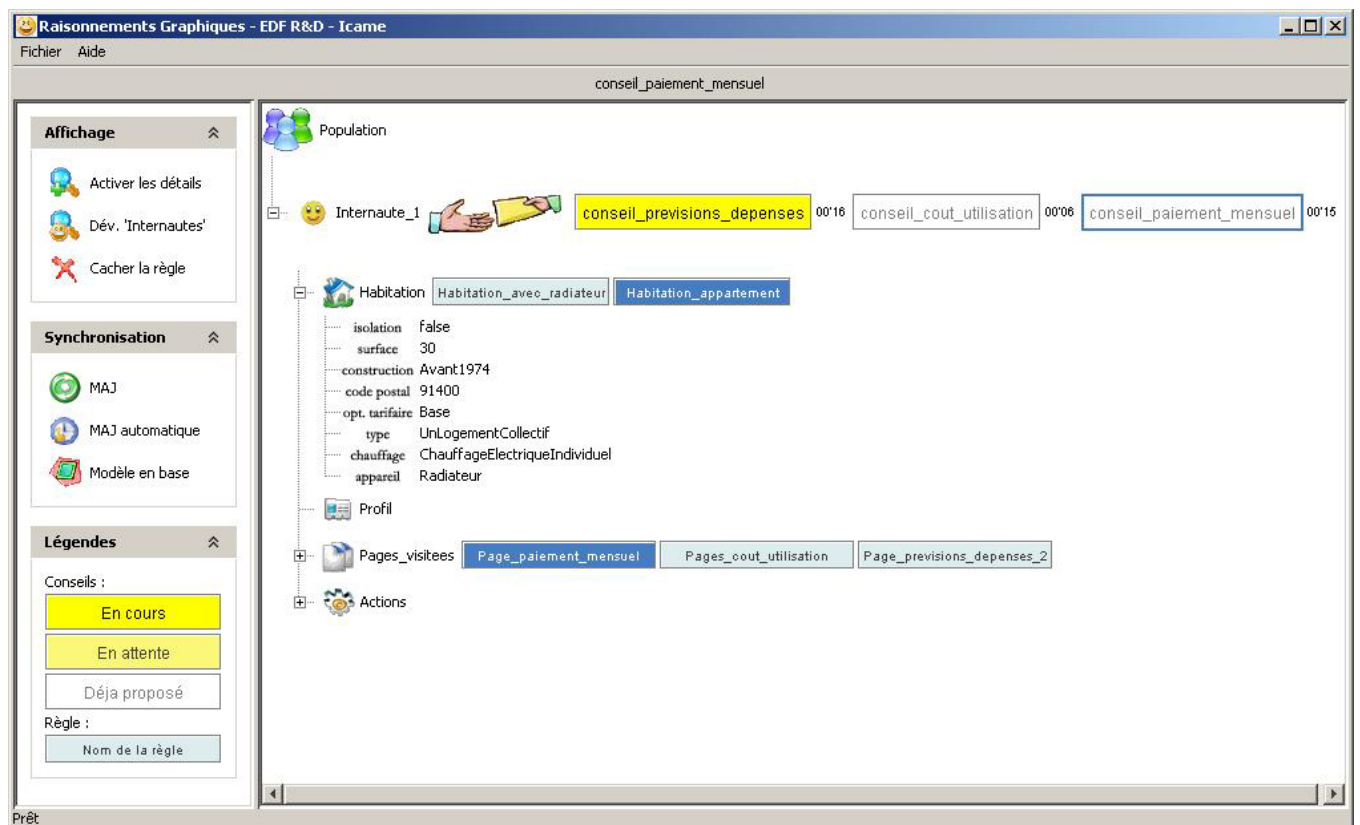


Figure 23 - Outil de visualisation du raisonnement (GRaison)

Dans cet exemple, l'internaute reçoit un conseil « conseil\_previsions\_dépenses » car son habitation est une « Habitation\_appartement » et que ses pages visitées inclues la page « Page\_paiement\_mensuel ».



L'explication va encore plus loin quand on demande le détail de la règle : nous avons la structure OWL intégrale de la règle explicitée et les informations concernant la logique de conseil : sa priorité, son titre, le texte à afficher, la page pointée, le conseil à enchaîner, ...

Figure 24 - GRaison : détails d'une règle

## 2. L'orgue de barbarie

Cet outil a été créé dans deux buts :

- le but de « datamining », connaître exactement ce qu'un internaute a fait sur le site et quelles informations il nous a laissé,
- le but de « tester » notre plateforme, « Quels conseils aurait été donné à cet internaute ? ».

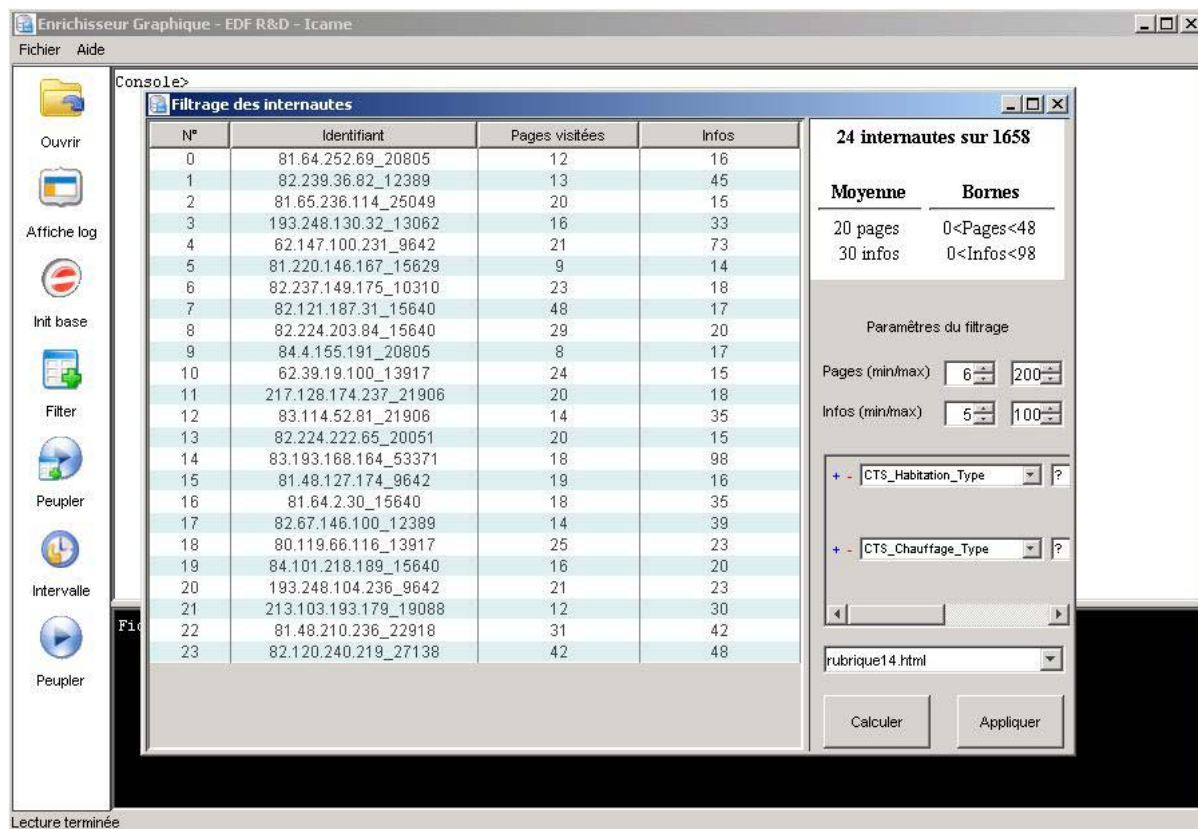


Figure 25 - Outil de datamining (GENrichis)

L'enrichisseur en orgue de Barbarie va alors lire le log ligne par ligne et réutiliser le composant veilleur pour le traitement de ces derniers. En parallèle, le raisonneur va nous afficher les conseils qui auraient été donné à l'internaute.

Afin de choisir les internautes les plus intéressants, des filtres peuvent être utilisés : nombre de pages vues minimum ou maximum, nombre d'informations données, page sur laquelle l'internaute doit passer, ...

Cet outil nous a particulièrement servi pour connaître :

- des parcours dans lesquels très peu de conseils étaient donnés,
- et expliquer le comportement des internautes qui ne visitaient que quelques pages et qui quittaient le site sans avoir trouver l'information qu'ils cherchaient.

### 3. La console d'administration

La console d'administration a été conçue pour gérer entièrement la plateforme. Il n'est alors plus nécessaire de modifier les fichiers de configuration et redémarrer le serveur.

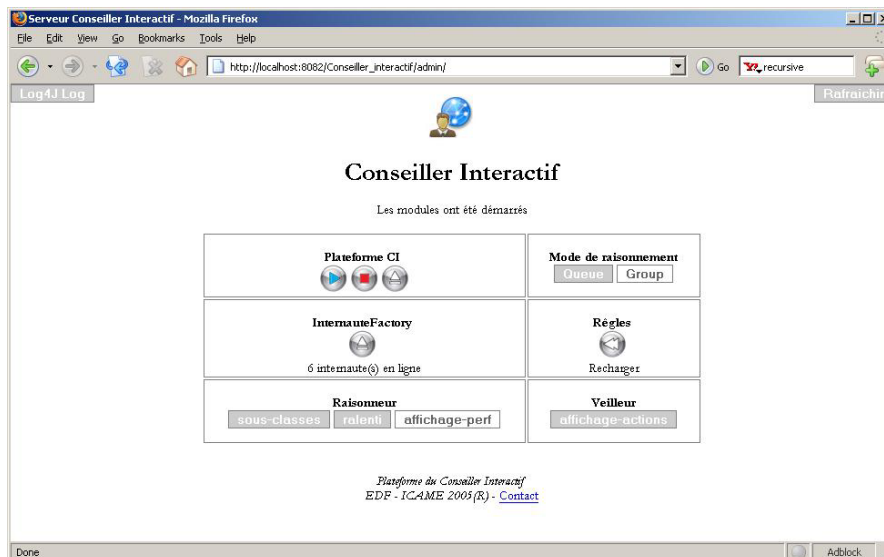


Figure 26 - La console d'administration

Il est également possible de gérer les internautes connectés : voir les informations, faire des réinitialisations ou désactiver le conseiller.

Des fonctionnalités de déboguages sont également disponibles : visualisation, sauvegarde et validation des modèles et autres ontologies.

Pour que la plateforme se rapproche le plus possible d'un modèle préindustriel, les fonctionnalités d'administration sont protégées (utilisation de Realm) et non accessibles par les internautes standards non authentifiés.



## K. L'architecture de la plateforme

Nous vous présentons ici l'infrastructure sur laquelle repose le prototype, le référentiel des données et l'architecture « finale » du « Conseiller Interactif ».

### 1. Infrastructure technique

Le prototype comporte quatre grandes parties : le **serveur Web** (existant), le **serveur d'application** qui va contenir toute la logique métier et effectuer les traitements, le **raisonneur OWL** qui un processus fonctionnant en parallèle qui se charge d'appliquer les règles marketing et la **base de données** qui capitalisent toutes les données « reçues » et « inférées » sur les internautes.

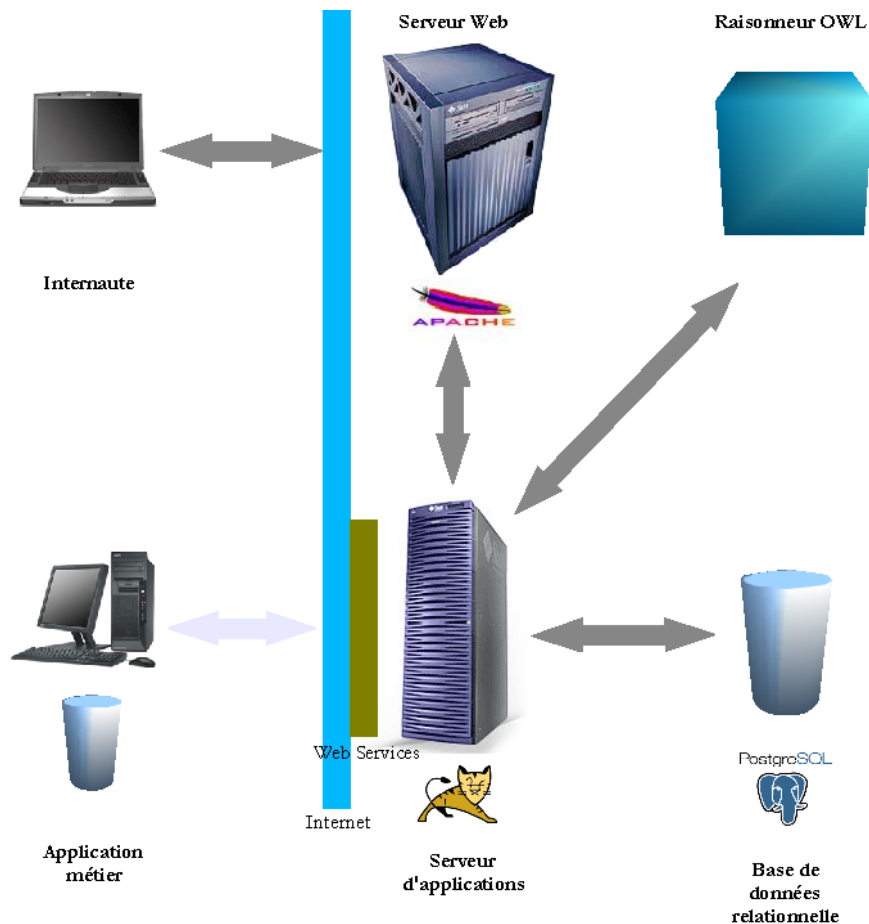


Figure 27 - Infrastructure technique du Conseiller Interactif

La couche de WebServices permet d'accéder aux données présentes dans la base et autorise une utilisation des services de raisonnements par des applications externes.

## 2. Technologies

Voici une vue ODP (Open Distributed Processing) montrant les différents modules du prototype : les technologies utilisées et les protocoles (ou méthodes) de communications.

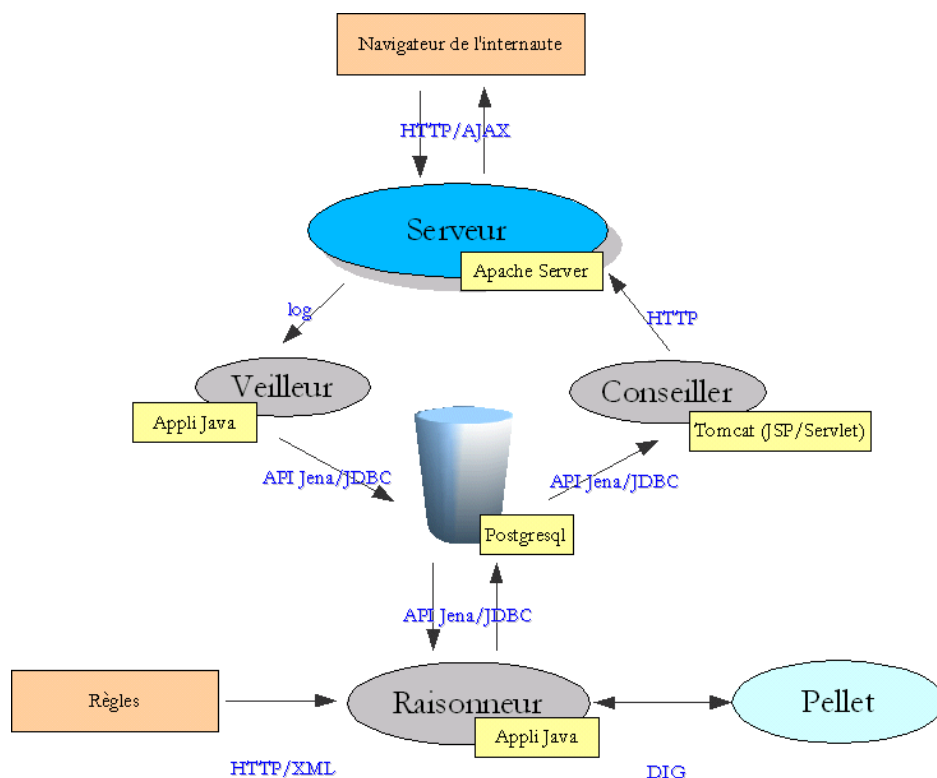


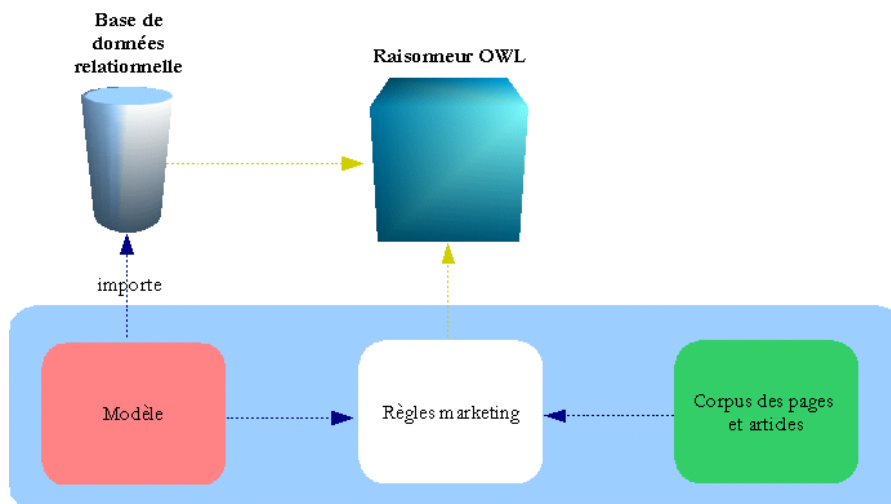
Figure 28 - ODP technologie

## 3. Architecture des données

La description des connaissances est aujourd'hui séparée des données proprement dites (informations brutes sur l'internaute) ; deux ontologies sont pour le moment utilisées : une ontologie « Règles marketing » qui englobe également l'ontologie décrivant le « modèle » internaute et une ontologie de pages et d'articles. L'objectif est le découplage intégral de l'infrastructure et de celle des connaissances :

Figure 29 - Référentiels

Chaque ontologie pourra alors avoir son propre cycle de vie, cela évitera à la plateforme un aspect monolithique et des utilisations éventuelles des ontologies dans des contextes différents.



#### 4. Architecture finale

Voici l'architecture finale du « Conseiller Interactif », les flux d'informations, les différents modules de traitement et de stockage et les différents acteurs intervenant pour le bon fonctionnement du système :

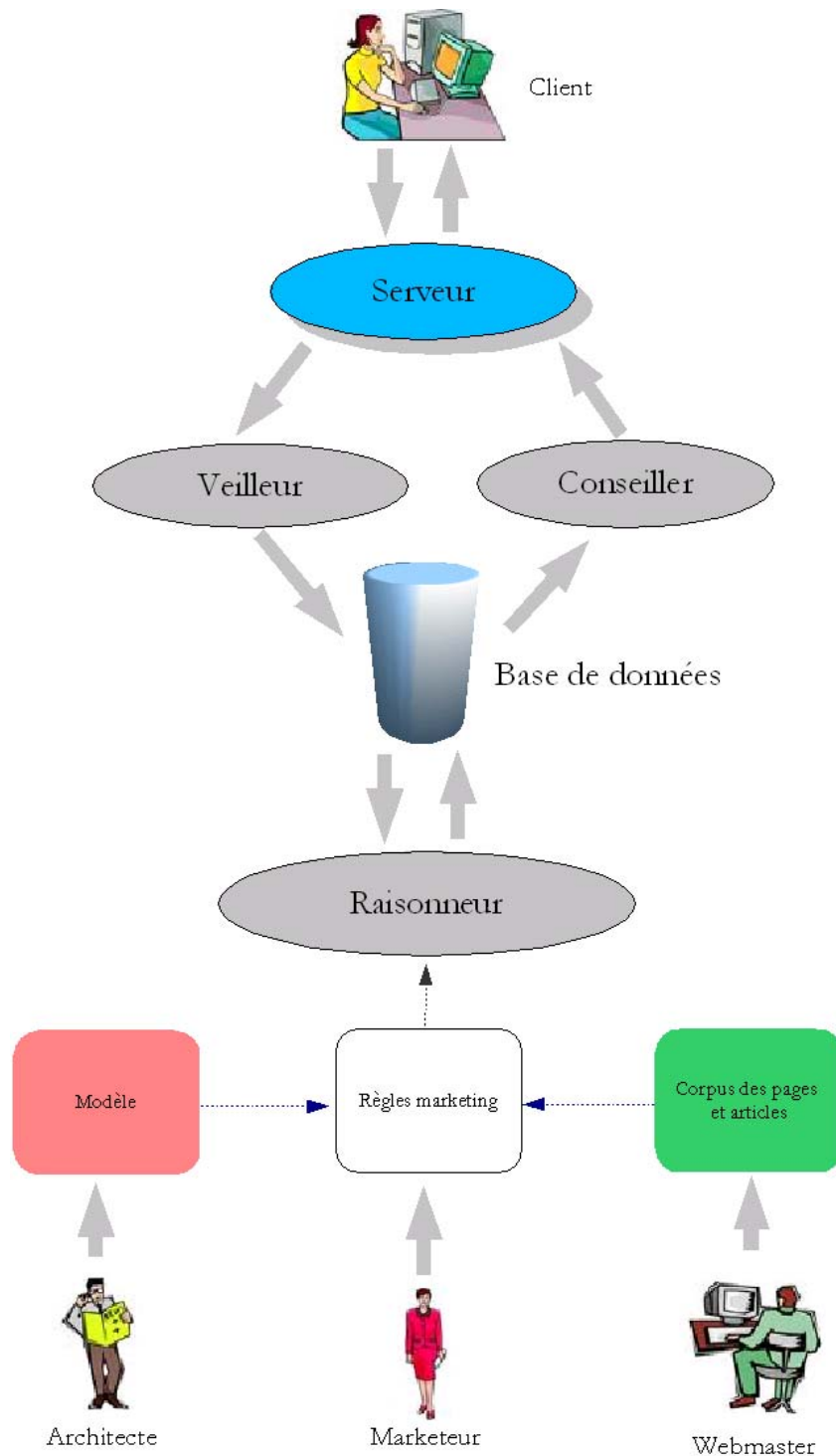


Figure 30 - Vue générale de la plateforme (ODP ingénierie)

## VI. Conclusion

### A. Bilan de la plateforme

L'objectif du Conseiller est la présentation d'un conseil dans des délais très courts : le temps de réponse ne devrait alors pas dépasser une seconde. Afin d'obtenir une plateforme robuste et avec un temps de réponse quasi-instantané, nous devons effectuer un audit de l'architecture dans son intégralité : les différents composants logiciels et autres modules, les « Frameworks » et également les choix d'implémentation.

#### 1. Mesure de performances

Nous allons ici examiner les diverses manières de mesure et d'optimisation du raisonneur.

##### a) Choix du raisonneur : Racer, RacerPro ou Pellet

Afin de sélectionner le raisonneur le plus approprié à notre plateforme, nous avons réalisé des « benchmark » afin de déterminer lequel était le plus véloce. Le programme effectuant le test simule la connexion *séquentielle* de deux internautes et leurs navigations ; chaque nouvelle page donne alors lieu à la réception d'un conseil.

Les tests nous donnent les résultats suivants :

- Racer 1.7 : 3.425 secondes
- RacerPro (1.8.1) : 3.35 secondes
- Pellet 1.3a : 3 secondes

A partir de ces résultats, on peut percevoir une avance de Pellet 1.3 sur les autres raisonneurs.

▶ Le raisonneur Pellet est plus véloce, il sera donc le raisonneur par défaut du prototype.

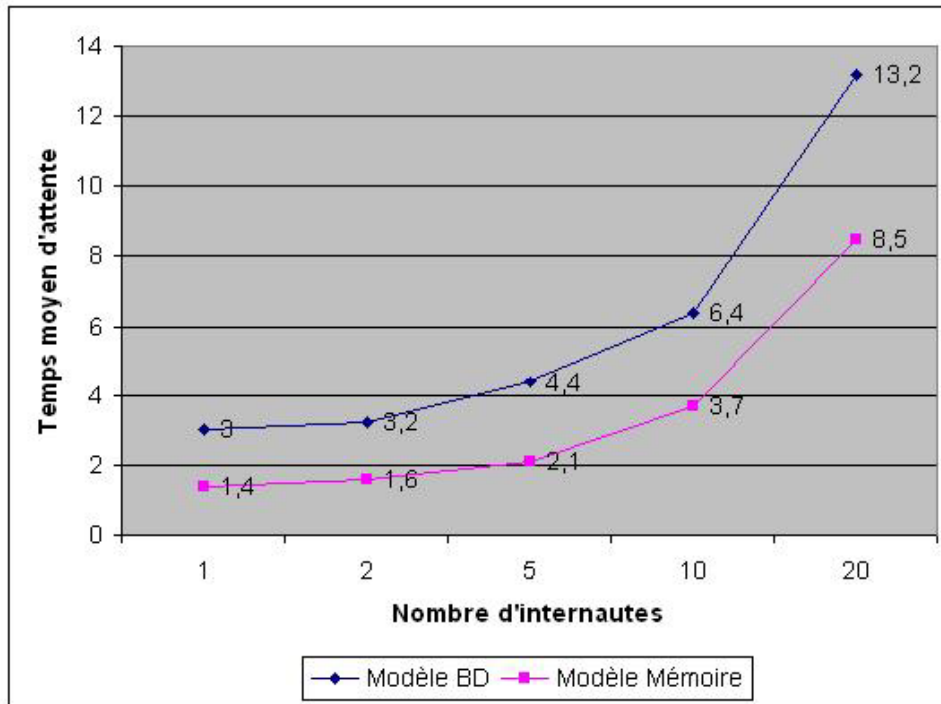
##### b) Modèle en base ou en mémoire ? Support multi-utilisateur ?

Les performances obtenues dans les tests précédents ne sont pas encore à la hauteur des attentes que l'on peut avoir pour une application Web (les temps de réponse ne doivent pas dépasser 2 secondes).

Nous optimisons le prototype en évitant l'utilisation d'une base et effectuons un test pour mesurer les temps de réponse pour une utilisation multi-utilisateur.

Nb Internaute	Modèle BD	Modèle Mémoire
1	3s	1,4s
2	3,2s	1,6s
5	4,4s	2,1s
10	6,4s	3,7s
20	13,2s	8,5s

L'utilisation d'un modèle en mémoire nous procure un gain significatif de performance, nous passons sous la barre symbolique des 2 secondes. Le prototype peut alors gérer 5 internautes avec des temps de réponse acceptables.



Toujours à la recherche de performances et en effectuant un « profiling » du code Java. J'ai remarqué que la majorité du temps de raisonnement (environ 60%) était consacrée au « binding » du modèle.

En voici l'explication : notre module raisonneur est connecté à un raisonneur OWL qui fait une copie de notre modèle, quand notre modèle change, celui du raisonneur OWL est réinitialisé (entièrement vidé puis rempli). De ce fait, avec l'augmentation du nombre d'informations et celui des internautes, le temps de raisonnement s'allongera d'une façon exponentielle.

La solution consiste alors à éviter le « binding » et mettre à jour « manuellement » le modèle du raisonneur OWL : le module *Connecteur\_Cache*.

### c) Utilisation d'un cache

Le connecteur cache est un modèle se mettant à l'écoute du modèle avec des « listeners » à la Java/Swing ; chaque fois que le modèle va changer, le cache va écouter et stocker les modifications (ajouts ou suppressions). Lorsque le raisonneur voudra mettre à jour son modèle, nous allons effectuer les ajouts et les suppressions plutôt que de recharger le modèle.

C'est au moment où le connecteur cache est implémenté (1<sup>er</sup> décembre 2005) que sort une nouvelle pre-release de Pellet : la version 1.3beta2-nb.

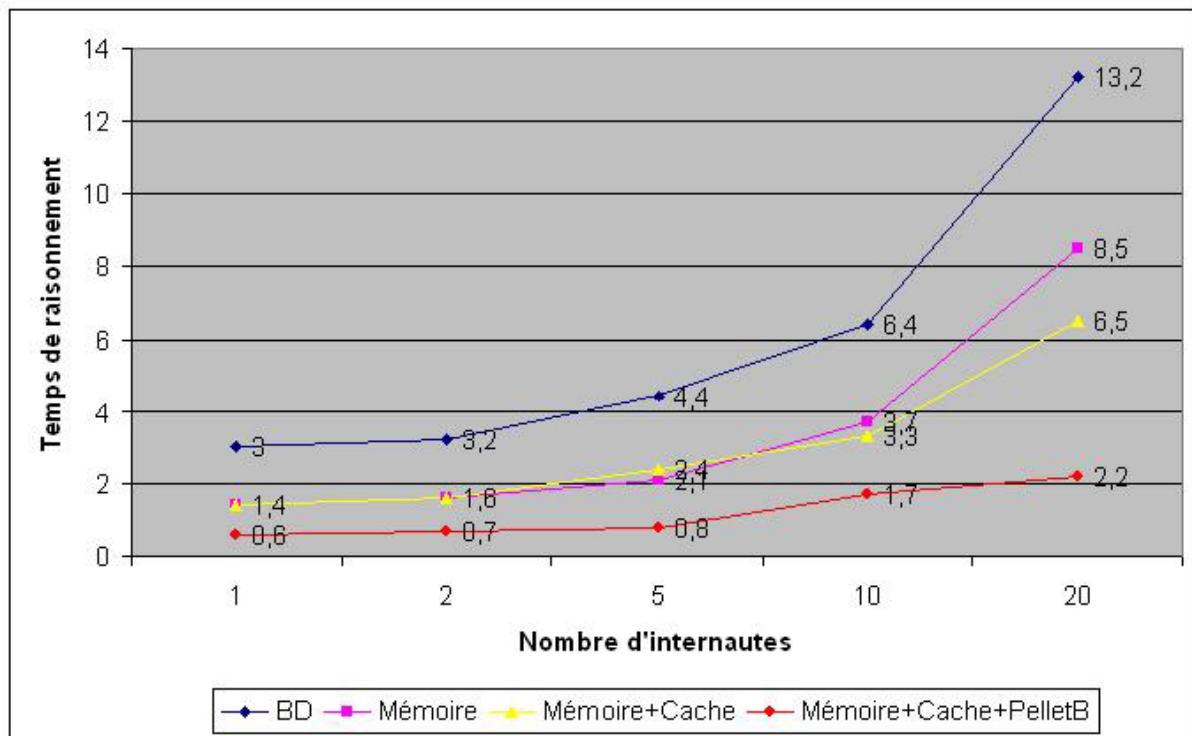
Cette version de Pellet est annoncée par ses créateurs comme le raisonneur OWL le plus rapide qui existe, j'effectue alors des mesures pour vérifier leurs dires.

Nb Internautes	Pellet 1.3 beta1	Pellet 1.3 beta2nb
1	1,4s	0,6s
2	1,6s	0,7s
5	2,4s	0,8s
10	3,3s	1,7s
20	6,5s	2,2s

Les résultats sont tout simplement spectaculaires : la combinaison du cache avec la dernière version de Pellet nous permet d'atteindre des temps de réponse très faibles. Ces temps malgré un nombre relativement important d'internautes restent faibles et constants contrairement aux précédentes mesures qui nous montraient des courbes évoluant exponentiellement.

► L'association d'un cache et de la dernière version Pellet nous fournit une plateforme rapide et fiable supportant jusqu'à 20 internautes.

Voici le tableau montrant les multiples évolutions du raisonneur et les gains de performance que ces modifications nous apportent :



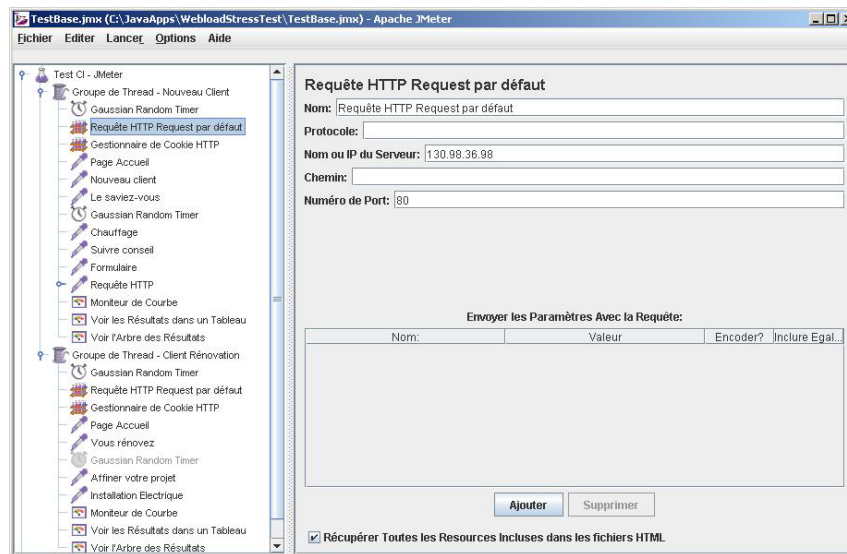
## 2. Etude de la scalabilité Web

La scalabilité est un critère important dans l'élaboration d'une plateforme : l'architecte doit prévoir les montées en charge et ainsi éviter de revoir toute l'architecture. Afin de contrôler le raisonneur et le support de plusieurs Internautes, nous avons recours au Framework JUnit<sup>53</sup>.

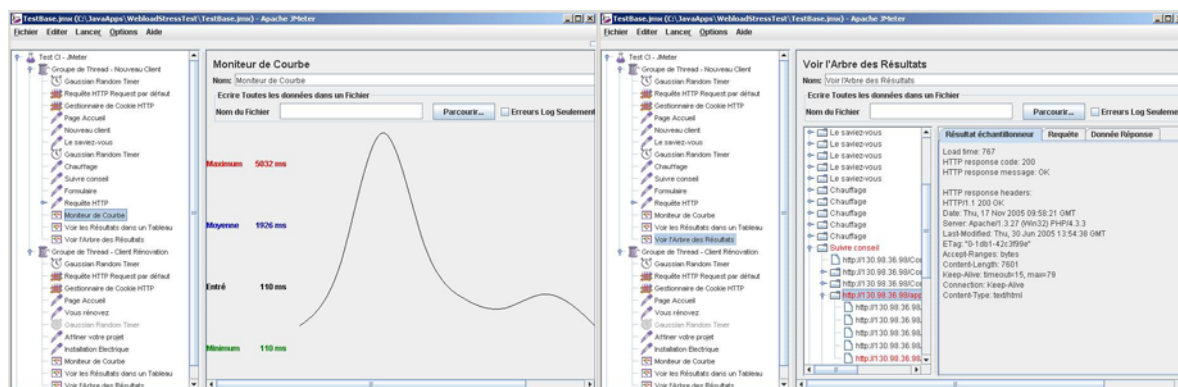
<sup>53</sup> <http://www.junit.org>

Nous allons simuler la connexion de plusieurs internautes et leurs navigations sur le site. Lors de la définition des « Test Case », nous établissons la liste des pages à visiter et les conseils correspondants. JUnit nous permet alors mesurer les temps de réponse et vérifier que l'on reçoit bien les « bons » conseils. Les résultats de scalabilité du raisonneur sont ceux établis dans la partie précédente.

Nous allons valider l'intégralité de la plateforme : le veilleur, le raisonneur, le serveur Web et les autres modules de gestion des internautes. Pour cela, nous utiliserons l'outil JMeter<sup>54</sup> de la fondation Jakarta. Cet outil dédié nous permet de décrire des scénarios de connexions (HTTP, HTTPS, SOAP) sur notre serveur Web et effectuer des tests et des mesures.



Nous simulons l'arrivée de 10 internautes et la navigation, l'outil supporte les Cookies et l'envoi de paramètres HTTP GET ou POST. A la fin du test, nous vérifions s'il n'y a eu aucune erreur, et pouvons calculer les courbes de charges du serveur.



Lorsque plusieurs internautes se connectent en même temps, il arrive que le serveur Apache n'arrive pas à correctement les identifier. De ce fait des internautes différents peuvent être amenés à recevoir les mêmes identifiants de Cookie.

<sup>54</sup> <http://jakarta.apache.org/jmeter/>

## ***B. Apports du Web Sémantique***<sup>55</sup>

Le Conseiller Interactif s'attaque au problème de l'intégration des connaissances dans un système de décision. En effet l'objectif est de décider en temps réel des conseils judicieux à prodiguer à l'internaute qui navigue sur le site Web des particuliers.

Le problème est d'avoir à la fois la souplesse de déclaration des connaissances de l'expert marketing pour spécifier les conseils et leurs conditions d'application et à la fois un système robuste qui intègre ces spécifications de la façon la plus transparente possible. Le Web Sémantique et principalement l'utilisation d'ontologies ont été choisis pour la richesse fonctionnelle et le large panorama d'outils disponibles dans ce monde.

### **1. Résolution du problème avec une ontologie**

#### **a) Définition d'une Ontologie**

Dans cette note on entend ici par ontologie une représentation formelle des connaissances et axiomatique de la logique, ce qui permet d'envisager de lui appliquer des raisonnements automatisables. L'ontologie est utilisée, depuis plusieurs années, dans l'ingénierie des connaissances pour structurer les concepts d'un domaine.

L'ontologie utilisée dans le cadre du prototype du Conseiller Interactif est basée sur le langage OWL. Ce langage est basé sur une logique de description particulière.

Les logiques de description sont des familles de logiques qui structurent l'expression des faits pour assurer notamment des propriétés de décidabilité des raisonnements logiques sur ces faits. Ainsi on retrouve dans les logiques de description les notions de concept, de rôle (soit une relation entre deux concepts soit un attribut d'un concept) et d'individu (une donnée qui représente un individu dont le type est un concept).

Ce qui change d'une logique de description à une autre sont les constructeurs logiques autorisés (il existe, quelque soit, contient au moins n relations de ce type, ...). Dans le cadre d'OWL un certain nombre de constructeurs permettent la déclaration par exemple de la généralisation entre concepts (notion de subsumption), la notion de restriction sur des relations en fonction du type associé dans la relation, la définition d'un concept comme l'union de deux concepts, comme le complémentaire d'un concept, etc.

Pour le langage OWL, trois niveaux de définition d'utilisation des constructeurs ont été définis :  
OWL-Lite : permet de décrire des ontologies simple mais manque de certains constructeurs logiques utilisés dans le Conseiller Interactif.

OWL-DL : permet une expressivité plus grande (union de deux concepts, rôle des plus d'un éléments) tout en étant assuré que le système logique reste décidable ce qui est vital pour le contrôle de cohérence.

OWL-Full : expressivité très grande : cela permet de considérer par exemple des attributs sur des concepts... le concept devient alors une instance... ce qui résulte en revanche dans le cas général à avoir un système logique indécidable.

---

<sup>55</sup> Extrait du REX sur le Web Sémantique – Laurent Olivry & Anthony Rabiata



Dans le cas du prototype du Conseiller Interactif, on utilise l'expressivité d'OWL-DL.

Ainsi l'ontologie est utilisée dans le Conseiller Interactif pour ses vertus de structuration de la pensée, à savoir la capacité de modélisation, et pour ses fondements logiques qui permettent d'automatiser des raisonnements.

### **b) La méthode de résolution avec une ontologie**

Pour résoudre le problème de déduction des conseils appropriés, il est nécessaire d'effectuer plusieurs étapes.

Dans un premier temps, il s'agit de **modéliser le problème**. La formalisation de cette modélisation sous la forme d'une ontologie est un exemple de mise en œuvre de cette modélisation. La modélisation du problème sous la forme d'une ontologie permet clairement de **spécifier des contraintes logiques (règles logiques)** au niveau du modèle.

Au niveau de la modélisation, on distingue trois sous niveaux :

Le **niveau stratégique** : il permet de déduire des concepts en fonction d'autres concepts notamment au niveau tactique (par exemple, on définit à ce niveau la pertinence de conseils à prodiguer en fonction d'autres concepts)

Le **niveau tactique** où l'on définit certains concepts de base en fonction des données collectées dans l'ontologie.

Le **niveau opérationnel** où l'on définit les données qui sont collectées en fonction de l'état du système.

Dans un deuxième temps, une représentation de cette ontologie est traduite dans le langage OWL, ce qui permet d'avoir une interface standardisée vers des moteurs d'inférence.

Dans un troisième temps, **un raisonnement** permet de déduire le sens des données par rapport à l'ontologie. Ce raisonnement se fait par l'exécution d'un moteur d'inférence (RACER ou PELLET).

En utilisant des moteurs d'inférences de type « chaînage arrière », leur appel nécessite de leur poser une question. En l'occurrence, pour le Conseiller Interactif, il s'agit de leur poser la question « De quel type est l'internaute ? ». Suite au raisonnement, la réponse peut-être « de type `Besoin_de_conseil_de_chauffage` »

Enfin, à l'aide des conclusions du moteur d'inférences, un conseil approprié, en l'occurrence dans l'exemple ci-dessus concernant le chauffage, est prodigué.

### **c) Les langages de requêtes**

Le Conseiller Interactif utilise l'expressivité de la logique de description sous-jacente à OWL pour exprimer les règles logiques et donc les règles marketing entre les différents concepts. Ainsi pour savoir quel est le conseil approprié pour un internaute n°1, le mécanisme consiste à poser la question au moteur d'inférence : « de quel type est l'internaute n°1 ».

La réponse à cette question peut se faire :

- grâce à l'expressivité de OWL et des règles logiques associées
- grâce à l'exécution d'un moteur de règles dédié (voir point suivant)
- grâce à l'utilisation d'une requête qui tiendrait compte de la structuration des connaissances... par exemple l'initiative SPARQL

Le langage de requête SPARQL est un langage qui permet d'interroger des fichiers RDF (Resource Description Framework). Comme OWL est un langage fondé sur RDF, il devient dès lors possible d'interroger l'OWL. La spécification de SPARQL est en cours de développement au W3C au sein du Data Access Working Group et en est au stade « Last Call Working Draft ».

Certains moteurs d'inférence se déclarent être en voie de conformité avec SPARQL comme le moteur KAON2 mais la maturité leur manque.

#### **d) Les langages de règles**

Le Conseiller Interactif utilise l'expressivité de la logique de description sous-jacente à OWL pour exprimer les règles logiques et donc les règles marketing entre les différents concepts.

Dans le cadre du Conseiller Interactif, nous n'avons pas utilisé de langage de règle car il n'existe pas à l'heure de langage de règle associé à OWL et standardisé par le W3C. Cependant la standardisation est en cours. Un groupe de travail nommé Rule Interchange Format (RIF) a commencé le 8 décembre 2005. Des candidats sont SWRL, RuleML mais le groupe de travail va peut-être préconiser un nouveau langage qui constituera un compromis.

Des limitations subsistent à l'élaboration d'un standard. Notamment le fait qu'il ne suffit pas de spécifier un langage de règles mais il faut également spécifier des règles d'inférences de moteurs. Car dans le cas contraire, à partir de mêmes jeux de règles il se pourrait avoir des déductions différentes.

L'utilisation d'un langage de règle peut permettre une adéquation plus fine entre l'expression de ces règles par un acteur métier et son codage de façon informatique.

Les règles de Horn de type « SI ... (conjonction de conditions) ALORS ... (conclusion) » sont très appropriées à l'expression de contraintes métier.

Nous avons effectué la mise en œuvre dans le prototype en respectant l'évolution des standards notamment du W3C. Ainsi, nous avons utilisé la couche la plus abstraite du Web sémantique : le langage OWL.

OWL permet d'exprimer un certain nombre de règles mais ne comporte pas réellement un langage de règles.

Une proposition nommé SWRL se fonde entièrement sur la sémantique d'OWL et permet d'exprimer des règles de Horn. Cette proposition contient quelques limites, notamment l'impossibilité d'exprimer des prédicats dont l'arité est supérieure à 2.

SWRL présente l'avantage de présenter une sémantique clairement définie. Ceci est un gage d'interopérabilité contrairement à RuleML qui ne présente que des cas tests d'utilisation.

Une seconde proposition est le langage RuleML, issu d'une initiative industrielle forte depuis 2000 pour tenter de rendre inter opérable des moteurs de règles... RuleML a pour objectif de fournir :

- un langage de règles standardisé,
- une plateforme inter opérable intégrant divers langages de règles métiers, des systèmes d'inférences et des paradigmes de représentation des connaissances.

RuleML travaille sur l'interopérabilité des standards tels que JSR 94, SQL'99, OCL, BPML, WSFL, XLang, XQuery, RQL, OWL, DAML-S, and ISO Prolog. RuleML travaille sur l'interopérabilité des systèmes tels que CLIPS, Jess, ILOG JRules, Blaze Advisor, Versata, MQWorkflow, BizTalk, Savvion, etc.

## 2. Bénéfices de l'utilisation d'une ontologie

Les ontologies permettent d'avoir une approche déclarative des connaissances d'un domaine. Ainsi l'expert métier exprimera dans un **vocabulaire très proche de son métier** les différentes connaissances qu'il aura à introduire dans le système de connaissance. Cette saisie se fera de façon déclarative et indépendamment des autres saisies potentielles d'autres branches de l'ontologie.

### a) Modularité

La modularité du prototype « Conseiller Interactif » s'exprime à la fois par l'utilisation de la représentation des connaissances sous forme d'ontologie mais également par des choix d'architecture applicative.

#### *(1) La représentation des connaissances par les ontologies*

L'expression de la connaissance sous la forme d'ontologies permet de séparer clairement les différents domaines de connaissance. Ainsi, deux branches différentes de la structure de l'ontologie pourront être maintenues en parallèle par des acteurs différents à des rythmes différents.

#### *(2) L'architecture applicative choisie*

Pour le Conseiller Interactif, l'architecture applicative a essayé d'utiliser des protocoles de communication les plus standardisés possible. En ce qui concerne les moteurs d'inférences compatibles avec les logiques de description notamment celle représentée par le langage OWL, il existe un protocole d'échange nommé DIG (Description Logic Implementation Group) qui est compatible avec plusieurs moteurs d'inférence. Ainsi, l'interchangeabilité des moteurs d'inférence offre une plus grande **pérennité** à l'architecture qui peut intégrer facilement un nouveau moteur compatible DIG.

L'utilisation de l'interface DIG permet d'échanger les moteurs d'inférences et de choisir le plus performant en terme de fonctionnalité et de rapidité. Le remplacement du moteur RACER par le moteur PELLET s'est traduit par une augmentation de 15 à 75% de rapidité dans les raisonnements.

### **b) Réutilisabilité**

Le prototype est construit pour permettre à des internautes sur un site Web de recevoir un certain nombre de conseils en fonction de leur profil. Ce profil est constitué de leur parcours sur le site Web ou bien des renseignements qu'ils fournissent lors de la saisie de formulaires.

La partie spécifique du prototype tient en la base de connaissance et la définition des conseils sous la forme de règles logiques. Cette partie est à reconstruire s'il s'agit de travailler sur un site Internet différent avec des pages différentes et des conseils différents.

En revanche la structure du « Conseiller Interactif » permettant d'héberger les connaissances, de veiller aux saisies et à la navigation de l'internaute, de prodiguer les conseils issus des raisonnements logiques est entièrement réutilisable.

### **c) Cohérence**

Les ontologies, basées sur une représentation axiomatique de la logique, offrent la possibilité de contrôler à tout moment l'ensemble de la **cohérence logique des concepts et contraintes logiques** déclarées.

L'utilisation d'une ontologie commune permet à divers acteurs de saisir des informations et de contrôler à l'aide des outils mathématiques mis à disposition a priori que l'ensemble des informations s'exprime en cohérence. Ainsi il est possible de **valider** une ontologie.

### **d) Maintenabilité**

La séparation entre modèle et logiciel est effective. La connaissance à proprement parlé est extraite du cœur du logiciel.

La séparation est nette entre la maintenance de l'ontologie qui est effectuée par un acteur métier et la maintenance du code informatique a proprement parlé.

### **e) Efficacité**

Le cœur du raisonnement est délégué au moteur d'inférence est laissé seulement à la charge du concepteur métier la description déclarative des relations logiques entre des concepts de l'ontologie.

L'outillage existe et fonctionne de façon optimisée, assurant ainsi une cohérence logique entre les données d'entrée et les conclusions de l'application.

### **f) Pérennité**

L'ontologie est construite à l'aide d'un outil (Protégé) qui permet de la spécifier dans le langage OWL. OWL est un standard du W3C sur lequel bon nombre d'outils émergent.

L'unification du langage d'expression d'ontologie est un atout en terme de **pérennité** car il est dès lors envisageable d'exécuter des raisonnements par plusieurs moteurs d'inférence différents sur la même ontologie.

Dans le cadre du prototype, l'expression des règles logiques s'est effectuée à l'aide de l'expressivité du langage OWL. Cette solution est un gage de **portabilité** entre différents moteurs d'inférence compatible OWL.

### **g) Traçabilité**

Les ontologies rendent les résultats d'un raisonnement explicables. A partir du moment où le moteur d'inférence produit l'explication de son raisonnement. Cette traçabilité est essentielle pour l'acceptation par l'expert de conclusions qui lui sembleraient abscons.

## **3. Conclusion sur le Web Sémantique**

Suite à l'élaboration et l'expérimentation du prototype, nous pouvons clairement conclure par une première analyse sur l'intérêt de l'utilisation d'une ontologie compte tenue des propriétés de modularité, réutilisabilité, cohérence, maintenabilité, efficacité, pérennité du prototype...

Tous ces critères confortent dans l'idée que l'utilisation d'une ontologie comme source de spécification fonctionnelle est un choix plus que technique. Il s'agit en fait de préparer, par l'utilisation d'un langage standardisé, des ponts futurs vers d'autres applications ; ces ponts étant fait au niveau sémantique et non au niveau technique. Les autres applications sont par exemple des applications fondées en partie sur l'ontologie du Conseiller Interactif mais étant fondées sur de nouvelles règles marketing ou autres... voire des applications similaires au Conseiller Interactif mais n'utilisant pas les même canaux de diffusion.

### ***C. Perspectives***

La mise en place de la plateforme du Conseiller Interactif nous a démontré la richesse fonctionnelle et les fortes capacités du Web Sémantique dans la GRC. L'objectif est maintenant de raffermir cette architecture afin de produire une version industrialisable.

Un projet nommé « Studio de services » dirigé par Pierre Chaussecourte va voir le jour début 2006. Son objectif est la réutilisation de tout le savoir faire que l'on a acquit avec le « Conseiller Interactif » dans des domaines similaires :

- dans le projet SEVE (Services, Conseils aux Clients Particuliers) pour les « préconisations automatiques »,
- dans le projet Stabilo pour la fourniture d'un « Conseil du jour ».

### ***D. Epilogue***

Participer à l'action « Conseiller Interactif » a été pour moi une expérience très enrichissante : sur le plan technique et sur le plan humain. Collaborer avec des spécialistes provenant d'univers différents m'a beaucoup plu et m'a permis de cerner les différentes exigences de chaque métier.

Cette mission m'a permis d'approfondir mes connaissances dans la GRC et d'appliquer mon savoir-faire dans les Nouvelles Technologies dans le cadre industriel d'EDF. J'ai ainsi acquis une réelle compétence dans les GRC et ceci m'a permis d'appréhender les enjeux qui y sont associés.

A la question : « Le Web Sémantique peut-il être appliqué pour de la GRC ? », nous pouvons alors répondre sans hésiter : « Oui, et d'une manière très élégante ». Assurément, le « Conseiller Interactif » répond à tous les besoins fixés en matière d'expressivité sur le profil client et en exigence marketing.

Les technologies du Web Sémantique ont atteint une maturité telle que les richesses théoriques qu'elles proposent peuvent enfin être utilisées dans un contexte industriel. Pour preuve, les moteurs d'inférences et autres outils du Web Sémantique sont maintenant adaptés à des utilisations « réelles ». La prochaine question qui se pose chez EDF est la suivante : « Peut-on appliquer ces technologies dans d'autres domaines intégrant une notion de connaissance ». Je vais tâcher d'y répondre de la même manière que pour le « Conseiller Interactif ».

## VII. Bibliographie

### a) Ouvrages et documents

La Gestion de la relation client – Ed Eyrolles 2005 – R. Lefébure, G. Venturi

Compte rendu de réunion : Organisation du projet Conseiller Interactif (20 janvier 2005), Pascale Capron, Myriam Louis Louisy, Pierre Chaussecourte, Stéphane Rouilly, Sophie Morin

Compte rendu de réunion : Note de cadrage de l'action Conseiller Interactif (22 avril 2005), Pascale Capron, Myriam Louis Louisy, Pierre Chaussecourte, Stéphane Rouilly, Sophie Morin

Le Conseiller Interactif : REX sur le Web Sémantique (Décembre 2005 - Janvier 2006), Laurent Olivry et Anthony Rabiaza

### b) Liens Internet

Le site Internet EDF

[www.edf.fr](http://www.edf.fr)

Le portail Intranet Icame

Vignette

<http://www.vignette.com/contentmanagement/0,2097,1-1-1928-4149-1968-4161,00.html>

Broadvision

<http://www.broadvision.com>

E-deal

<http://www.e-deal.com/software/automation.jsp>

Liveperson

<http://www.liveperson.com/sb>

CraftySyntax

<http://www.craftysyntax.com>

Entreprise Daumas, Optimisation des ressources et des processus

<http://www.daumas-autheman.com>

Intégration d'un moteur de règles

<http://www.01net.com/article/256828.html>

JRules

<http://www.ilog.fr/products/jrules/>

Integration of Business Rules Engine in J2EE Server

[http://dev2dev.bea.com/pub/a/2005/04/business\\_rules\\_engines.html](http://dev2dev.bea.com/pub/a/2005/04/business_rules_engines.html)

Jess

<http://herzberg.ca.sandia.gov/jess/>

Drools

<http://drools.org/>

JSR 94

<http://www.jcp.org/en/jsr/detail?id=94>

XML

<http://www.w3c.org/XML/>

RDF

<http://www.w3c.org/RDF/>

OWL

<http://www.w3c.org/OWL/>

RDF OWL Toolkits

<http://www.wiwiss.fu-berlin.de/suhl/bizer/toolkits/#proprietary>

Jena

<http://jena.sourceforge.net>

Joseki

<http://www.joseki.org>

Protégé

<http://protege.stanford.edu/>

<http://protege.cim3.net/cgi-bin/wiki.pl>

OilEd

<http://oiled.man.ac.uk/>

Altova Semantic Works

[http://www.altova.com/products\\_semanticworks.html](http://www.altova.com/products_semanticworks.html)

Racer

[www.cs.concordia.ca/~haarslev/racer](http://www.cs.concordia.ca/~haarslev/racer)

RacerPro

<http://www.racer-systems.com/>

Pellet

<http://www.mindswap.org/2003/pellet/>

DIG

<http://dl.kr.org/dig/>