

---

# Utilisation d'Ontologies pour la Coordination dans le Workflow Inter-Organisationnel Lâche

Lotfi Bouzguenda\* — Rafik Bouaziz\* — Eric Andonoff\*\*

\*Laboratoire MIRACL

ISIMS, route de Tunis, km 10, Sfax, Tunisie

Lotfi.Bouzguenda@isimsf.rnu.tn, Rafik.Bouaziz@fsegs.rnu.tn

\*\*Laboratoire IRIT

UTI, 2 rue du Doyen Gabriel Marty, 31042 Toulouse Cédex, France

Eric.Andonoff@univ-tlse1.fr

---

**RÉSUMÉ.** Cet article montre comment on peut utiliser les ontologies pour traiter le problème de la coordination de processus workflow dans le contexte du Workflow Inter-Organisationnel (WIO) lâche. Le WIO a pour objectif de faire coopérer des processus workflow issus de plusieurs organisations qui sont par nature hétérogènes, autonomes et réparties afin de produire un nouveau service. Le WIO lâche correspond à une coopération occasionnelle, faiblement structurée, entre des organisations non nécessairement connues a priori. Les principaux problèmes de coordination dans le WIO lâche sont : (i) la recherche de partenaires (organisations) capables de mettre en oeuvre des services workflow (c'est-à-dire de services implémentant des processus workflow) et (ii) la négociation de ces services entre partenaires. Cet article montre comment deux infrastructures de médiation, définies pour traiter ces problèmes de recherche et de négociation, utilisent des ontologies qui aident à la découverte automatique de services workflow et au choix d'un protocole de négociation.

**ABSTRACT.** This paper shows how it is possible to use ontologies to deal with coordination of workflow processes in the loose Inter-Organizational Workflow (IOW) context. The aim of IOW is to support the cooperation between distributed and heterogeneous business processes running in different autonomous organizations to reach a common goal, corresponding to a value-added service. Loose IOW refers to occasional cooperation, free of structural constraints, where the involved partners (organisations) and their number are not pre defined. Coordination in loose IOW raises several specific problems which are: (i) finding partners able to realize a workflow service (i.e. a service implementing a workflow process), (ii) the negotiation of workflow services between partners. This paper shows how two specific components introduced to deal with finding partners and negotiation between partners use ontologies for automatic discovering of workflow services and for choosing a negotiation protocol.

**MOTS-CLÉS:** Workflow Inter-Organisationnel, Coordination, Ontologie.

**KEYWORDS:** Inter-Organizational Workflow, Coordination, Ontology.

---

## 1. Introduction

Les ontologies sont utilisées dans divers domaines de l'informatique tels que l'ingénierie des connaissances, les systèmes d'information, les systèmes multi-agents ou le Web sémantique. Elles ont pris aujourd'hui une importance considérable car elles permettent de capitaliser et partager des connaissances pour résoudre des problèmes d'hétérogénéité sémantique qui existent inévitablement dans les applications distribuées issues de ces domaines. Les ontologies sont utilisées pour la recherche d'information (Hernandez, 2005), l'enseignement assisté par ordinateur (Lando, 2006), la découverte et la composition de services Web (Medjahed *et al.*, 2003) ou encore comme support pour l'automatisation de la négociation dans les systèmes multi-agents (Tamma *et al.*, 2005).

Le Workflow Inter-Organisationnel (WIO) peut lui aussi tirer profit des ontologies. Le WIO est une extension du Workflow traditionnel. Il a pour objectif de coordonner des processus workflow issus de plusieurs organisations qui sont par nature hétérogènes, autonomes et réparties. Cette hétérogénéité se manifeste par le fait qu'une même connaissance peut être représentée par différents termes dans des organisations différentes ou bien par le fait qu'un même terme désigne des connaissances différentes.

Nous avons étudié le problème de la coordination de ces processus hétérogènes, autonomes et répartis. Ce problème est central pour le WIO et les solutions proposées dans la littérature sont inadaptées au contexte dynamique dans lequel Internet nous a plongé (Bouzuenda, 2006).

Deux scénarios, mis en évidence dans (Divitini *et al.*, 2001), permettent de classer les applications de type WIO : un scénario lâche et un scénario serré. Le WIO lâche correspond à une coopération occasionnelle et faiblement (ou non) structurée entre des organisations non nécessairement connues à priori tandis que le WIO serré fait intervenir des organisations connues, avec lesquelles la coopération est souvent expérimentée et est fortement structurée.

Nous avons étudié le problème de la coordination de processus dans le cadre du WIO lâche et avons identifié deux principaux sous problèmes à résoudre pour coordonner de manière efficace des processus workflow impliqués dans un tel contexte, qui, par nature, est dynamique et ouvert puisque les partenaires avec lesquels la coopération se fait peuvent changer y compris lors de l'exécution (Bouzuenda, 2006) :

- La *recherche de partenaires* qui permet de connecter une organisation cherchant de l'aide pour réaliser un service workflow (demande), c'est-à-dire un service implémentant un processus workflow, à une ou plusieurs organisations capables de mettre en œuvre le service recherché (offre).
- La *négociation entre partenaires*, dont l'objectif est de permettre de choisir, parmi les partenaires sélectionnés lors de l'étape précédente, celui qui va mettre en

œuvre le service workflow recherché. L'organisation demandant de l'aide évalue et négocie les différents services workflow proposés selon différents critères tels que, le prix du service, la date de réalisation du service, la visibilité de l'évolution du service en cours d'exécution ou encore la manière de le mettre en œuvre.

Nous avons défini et mis en œuvre une architecture pour la coordination dans le WIO lâche. Plus précisément, cette architecture permet à différents Systèmes de Gestion de Workflow (SGWf) appartenant à différentes organisations impliquées dans un WIO lâche de se coordonner pour exécuter les différents processus workflow participant au processus workflow global (processus WIO). Chaque SGWf est également capable de se connecter à deux infrastructures de médiation définies pour traiter les deux sous problèmes de coordination listés précédemment, à savoir la recherche de partenaires et la négociation entre partenaires (Andonoff *et al.*, 2006).

Ces deux infrastructures intègrent des composants logiciels implémentant des protocoles de coordination (plus précisément des protocoles supportant la recherche de partenaires et la négociation entre partenaires) et exploitant :

- une *ontologie de domaine* qui permet de fixer le vocabulaire du processus WIO et qui aide par conséquent à la résolution des conflits d'ordre sémantique, souvent inévitables dans un tel contexte,
- une *ontologie de négociation* qui définit les (types de) protocoles utilisables lors d'une négociation dans le contexte du WIO lâche.

L'ontologie de domaine est utilisée lors de la phase de recherche de partenaires. Elle sert en premier lieu de support à la spécification des offres et des demandes de services workflow. Elle facilite ensuite la découverte automatique de services workflow puisqu'elle permet d'affiner le processus de recherche de partenaires qui met en correspondance une demande et des offres de services. Le recours à cette ontologie permet l'implémentation de mécanismes de filtrage (comparaison) entre une demande et des offres qui mettent en œuvre autre chose qu'une simple égalité.

Quant à l'ontologie de négociation, elle simplifie tout d'abord la conception du processus WIO par le fait qu'elle permet le partage de l'ensemble des protocoles de négociation qui ne sont ainsi plus codés dans l'ensemble des processus workflow impliqués dans un WIO. Cette ontologie permet ensuite le choix dynamique du protocole sous jacent à une négociation : ce protocole va garantir la cohérence sémantique des interactions entre les partenaires impliqués dans un WIO.

Cet article a pour objectif de montrer comment il est possible d'utiliser ces deux ontologies lors des phases de recherche de partenaires et de négociation entre partenaires. Il est organisé comme suit. La section 2 présente l'architecture de coordination que nous proposons pour le WIO lâche. Plus précisément, cette section montre comment nous modifions l'architecture du SGWf pour que ce dernier soit capable de se connecter aux deux infrastructures de médiation définies pour la recherche de partenaires et la négociation entre partenaires. Les sections 3 et 4 sont respectivement consacrées à ces deux infrastructures. Elles montrent notamment

comment les infrastructures utilisent les ontologies, d'une part pour aider aux comparaisons entre les offres et les demandes de services workflow, et d'autre part pour aider au choix du protocole utilisé lors de la phase de négociation. La section 5 situe notre travail par rapport à l'état de l'art et conclut l'article en traçant quelques perspectives de travaux à réaliser.

## **2. Une Architecture de Coordination pour le WIO lâche**

Nous avons spécifié une architecture pour la coordination de services workflow impliqués dans un WIO lâche (Andonoff *et al.*, 2006). Cette architecture comprend trois composants principaux : un (ou plusieurs) Système(s) de Gestion de Workflow (SGWF), et une infrastructure de médiation supportant la recherche de partenaires et une infrastructure de médiation supportant la négociation entre partenaires. Les deux infrastructures sont utilisées et partagées par les SGWfs des différentes organisations impliquées lors de l'exécution d'un processus WIO.

Cette section présente l'architecture que nous proposons pour qu'un SGWF soit capable de participer à l'exécution d'un processus de WIO lâche. Les sections 3 et 4 sont, quant à elles, consacrées à la présentation des infrastructures de médiation ; elles détaillent notamment le rôle joué par les ontologies de domaine et de négociation.

Cette architecture a pour point de départ l'architecture de référence définie par la Workflow Management Coalition (WfMC, 1994) qui est un organisme visant à la standardisation de tout ce qui est relatif au workflow. Tout comme (Buhler *et al.*, 2005), nous pensons que, bien qu'ancienne, cette architecture de référence définit un cadre cohérent et actuel pour parler des SGWfs.

Le composant principal défini par cette architecture de référence est le moteur de workflow. Son rôle est d'assurer l'exécution des processus workflow et de supporter les cinq interfaces (appelées WAPI pour Workflow API) suivantes :

- Interface 1 qui fait le lien entre le moteur et le composant supportant la définition des schémas de processus,
- Interface 2 qui fait le lien entre le moteur et les composants développés pour supporter les interactions avec les utilisateurs (acteurs jouant un rôle dans l'exécution des processus),
- Interface 3 qui fait le lien entre le moteur et les composants représentant des applications externes sollicitées lors de l'exécution des processus (serveur de bases de données, annuaires Ldap...),
- Interface 4 qui fait le lien entre différents moteurs de workflow,
- Interface 5 qui fait le lien entre le moteur et le composant supportant le paramétrage et la surveillance de l'exécution des processus workflow.

Notre proposition reprend la conception du moteur de workflow en l'adaptant au contexte du WIO lâche tout en conservant les cinq interfaces, notamment l'interface 4 qui va être particulièrement utilisée lors de l'exécution du processus WIO distribuée sur plusieurs SGWf. Plus précisément, nous avons agentifié le moteur de workflow et défini un agent spécifique pour la connexion vers l'extérieur (autres moteurs de workflow et infrastructures de médiation).

Notons que l'objectif de l'article n'est pas de défendre l'intérêt de l'approche agent dans le contexte du WIO lâche. Le lecteur intéressé peut, pour cela, consulter (Buhler *et al.*, 2005) et (Bouzguenda, 2006). Cependant, voici brièvement quatre arguments fondamentaux qui justifient l'agentification du moteur de workflow :

- Tout d'abord, l'approche agent fournit des abstractions et des protocoles de haut niveau pour traiter l'autonomie, la distribution et l'hétérogénéité qui sont inhérentes au WIO lâche : chaque SGWf local impliqué dans un WIO est vu comme une entité représentant une organisation, et responsable d'une mission qu'elle est capable d'accomplir en coopération avec d'autres entités.

- Ensuite, l'approche agent permet de décrire la coordination d'un WIO lâche, tant au niveau micro à travers les agents eux-mêmes, qu'au niveau macro à travers des abstractions organisationnelles telles que les rôles, les groupes, les engagements...

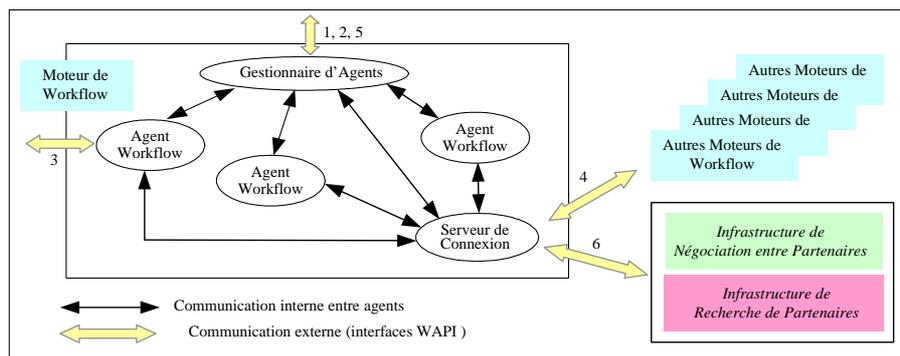
- L'approche agent est également adéquate pour modéliser et exécuter des processus adaptatifs. En effet, dans le contexte du WIO lâche, il est nécessaire d'être capable de modifier dynamiquement le processus WIO en cours d'exécution suite à la défection d'un partenaire : on est alors amené à modifier le processus WIO afin de supporter la recherche et la négociation entre partenaires en introduisant des séquences de patterns (patrons) workflow.

- Enfin, la coordination dans les systèmes multi agents est un thème à part entière, et l'on peut exploiter les solutions apportées dans ce contexte pour traiter le cas particulier du WIO lâche.

L'architecture du moteur de workflow, présentée en figure 1, inclut : (i) un *Agent Workflow* pour chaque instance de processus workflow en cours d'exécution, (ii) un *Gestionnaire d'Agents* chargé de gérer ces agents, (iii) un *Serveur de Connexion* qui interagit avec l'extérieur, notamment les autres moteurs de workflow et les deux infrastructures de médiation spécialisées pour la recherche de partenaires et la négociation entre partenaires, et (iv) une nouvelle interface, *l'Interface 6*, supportant la communication entre le serveur de connexion et les infrastructures de médiation.

En ce qui concerne les agents workflow, l'idée est d'implémenter chaque instance de processus workflow (cas) comme un processus logiciel, et d'encapsuler ce processus dans un agent actif. Un tel agent workflow inclut un moteur workflow qui lit la définition du processus workflow et déclenche les actions qui doivent être conduites en fonction de son état. Plus précisément, chaque agent workflow inclut sa propre définition du processus (ce qu'il doit exécuter lui est propre), et chaque

agent inclut son propre moteur pour interpréter cette définition (comment l'exécuter lui est également propre). Ainsi, l'agent workflow peut plus facilement s'adapter aux demandes de changement qui sont inhérentes au contexte du WIO tâche. Enfin, cet agent workflow supporte l'interface 3 de façon à solliciter les applications nécessaires à la mise en œuvre des tâches du processus workflow auquel il correspond.



**Figure 1.** Architecture Agent pour le Moteur de Workflow

Le gestionnaire d'agents contrôle et coordonne le fonctionnement des agents workflow :

- A partir d'une demande de création d'une nouvelle instance de processus workflow, il crée une nouvelle instance de l'agent workflow correspondant, initialise ses paramètres (son identifiant, sa date de création, sa durée de vie, etc.) en fonction du contexte (exécution d'une instance de processus workflow, négociation ou publication d'une demande dans le cas d'un agent workflow demandeur, ...), et lance son moteur workflow.

- Il assure la persistance des agents workflow qui exécutent des processus d'organisations à long terme et dans laquelle le traitement des tâches s'entrelace avec des périodes d'inactivité.

- Il coordonne les agents workflow dans l'utilisation des ressources partagées.

- Il fait le lien avec l'environnement via les interfaces 1, 2 et 5.

- Il permet également, dans le cas d'un SGWf fournisseur, de publier des offres et de négocier des services workflow implémentant des processus workflow.

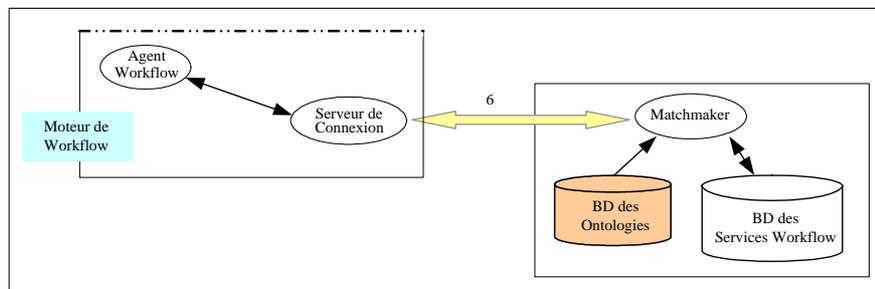
Le rôle du serveur de connexion est d'aider les agents workflow à se connecter vers l'extérieur pour soit solliciter l'exécution d'un service workflow auprès d'un autre moteur de workflow (via l'interface 4), soit pour trouver les partenaires dont ils ont besoin. En ce qui concerne ce dernier point, le serveur de connexion interagit avec les infrastructures de médiation spécialisées pour la recherche de nouveaux partenaires ou la négociation entre partenaires. Cette interaction exige une nouvelle interface, l'interface 6, qui permet la communication entre le serveur de connexion

et les infrastructures de médiation, et par conséquent entre différents serveurs de connexion de différents moteurs de workflow.

### 3. Utilisation de l'Ontologie de Domaine lors de la Recherche de Partenaires

Cette section détaille dans un premier temps l'infrastructure de recherche de partenaires, puis montre ensuite comment nous utilisons l'ontologie de domaine pour la comparaison d'offres et de demandes de services workflow.

L'infrastructure de médiation proposée supporte la recherche de partenaires capables de réaliser un service workflow (cf. figure 2). Elle se connecte dynamiquement au moteur de workflow via le serveur de connexion et l'interface 6.



**Figure 2.** Infrastructure de médiation pour la Recherche de Partenaires

Cette infrastructure est organisée autour des composants suivants :

- un agent *Matchmaker* qui permet de comparer une demande et des offres déjà publiées et de retourner l'identité d'un (ou plusieurs) fournisseur(s) capable(s) de répondre aux besoins exprimés par le demandeur,
- une base de données *Services Workflow* qui stocke les services workflow publiés (offerts) par des organisations fournisseurs,
- une base d'*Ontologies* qui stocke des ontologies de divers domaines.

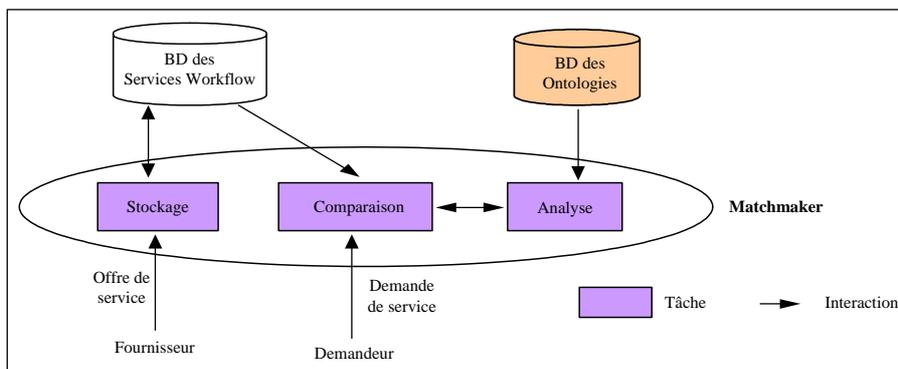
Comme illustré en figure 3, l'activité de l'agent *Matchmaker* est composée des trois tâches décrites ci-après.

La tâche de Stockage consiste à enregistrer les services workflow publiés par les fournisseurs dans la base *Services Workflow*. L'ontologie de domaine sert de base à cette spécification car seul le vocabulaire spécifié dans cette ontologie peut être utilisé d'une part pour définir les informations et les acteurs (rôles) dont un service workflow a besoin pour être réalisé, et d'autre part pour définir les informations que ce service produit en retour. Les services workflow sont spécifiés en utilisant de manière conjointe les Réseaux de Pétri à Objets (Sibertin, 1985) et le langage de description de services web sémantique OWL-S. Le lecteur intéressé trouvera dans

(Andonoff *et al.*, 2005) des éléments explicitant cette spécification de services workflow.

La tâche d'Analyse a pour point de départ l'ontologie de domaine exprimée au format OWL. Dans (Bouzuenda, 2006), nous avons défini une étude de cas appelée «Relecture des articles soumis à une conférence» qui relève du WIO tâche. Plus précisément, nous avons spécifié l'ontologie de domaine «Conference» qui décrit de manière hiérarchique l'ensemble du vocabulaire employé dans le domaine de l'étude de cas. Cette ontologie a été spécifiée à l'aide de Protégé-3.2.1 (Protégé-2000) et a été opérationnalisée au format OWL à l'aide d'un Plug-In spécifique(cf. figure 4).

La tâche de Comparaison, activée par une demande de service workflow, sélectionne dans la base de données les (offres de) services workflow compatibles avec cette demande.



**Figure 3.** Architecture de l'Agent Matchmaker

Plus précisément, la tâche d'analyse sélectionne l'ontologie de domaine correspondant à la demande, en extrait les classes et leurs liens et construit l'arborescence correspondante. Dans notre contexte, cette action est possible puisque le vocabulaire défini dans l'ontologie de domaine est décrit sous forme hiérarchique. Chaque sommet de cette arborescence correspond à une classe de l'ontologie et chaque arc correspond à une relation de sous-classe. Cette arborescence permet de déduire des relations de généralisation (subsumption) entre les concepts, c'est-à-dire le fait qu'un concept soit plus général qu'un autre. Un concept C englobe (subsume) un concept C' si l'extension de C' est incluse dans celle de C. On dira alors que C est plus général que (ou englobe) C'. Ce principe nous permet de réaliser des comparaisons flexibles entre les offres et les demandes, c'est-à-dire d'associer à une demande des offres qui ne correspondent pas exactement aux besoins exprimés mais qui s'en rapprochent.

Pour illustrer notre propos, prenons un exemple issu de l'ontologie «Conference» que nous avons défini dans l'étude de cas «Relecture des articles

soumis à une conférence». Dans cette ontologie (cf. Figure 4), le concept *Domaine* englobe le concept *Groupware* qui lui-même englobe les concepts *Workflow* et *EDI*. Ainsi, si l'on dispose d'une part d'un lecteur capable de lire un article dans le domaine de Groupware, et d'autre part, d'un demandeur à la recherche d'un lecteur d'article dans le domaine de Workflow, il est possible de les mettre en contact même si leurs déclaration (de capacités) ne sont pas exactement identiques, dans la mesure où l'on considère que le domaine du Groupware englobe celui du Workflow.

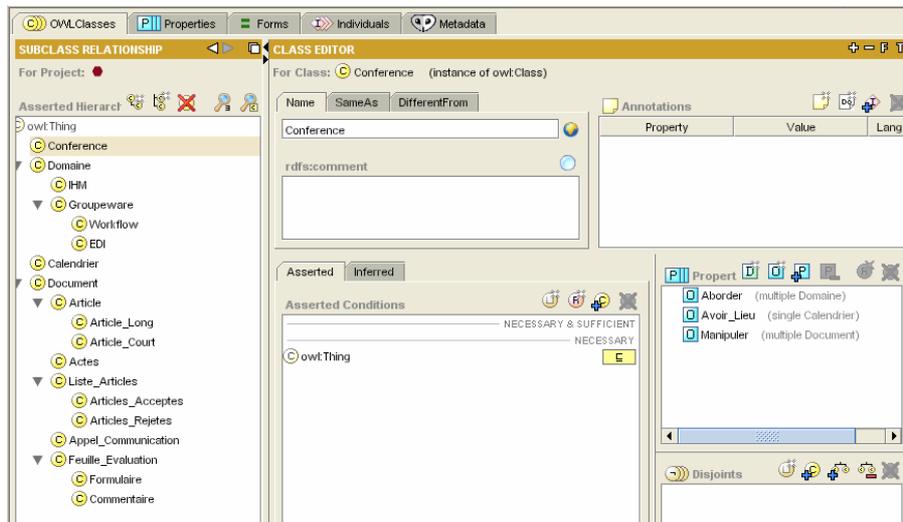


Figure 4. Extrait de l'Ontologie

Le résultat de la tâche d'analyse est partiellement décrit en figure 5. Cette figure visualise à la fois l'arborescence issue du concept *Domaine* et le fichier OWL qui sert de point de départ à la construction de cette arborescence.

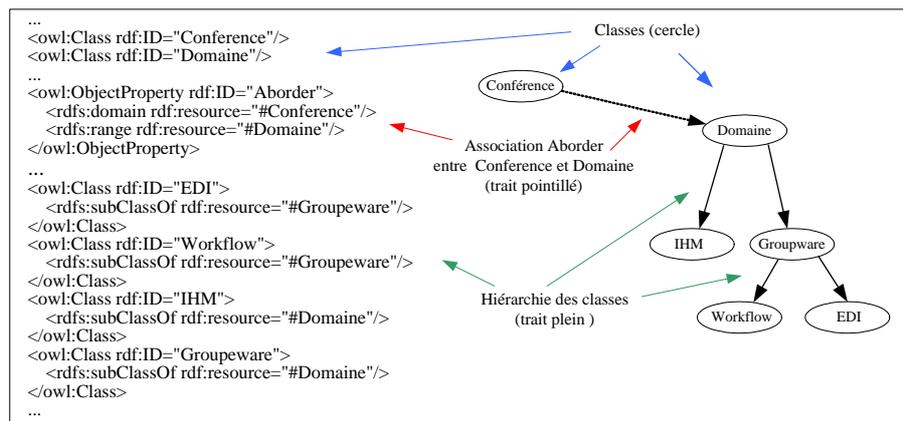


Figure 5. Extrait de l'Ontologie

La tâche de comparaison permet donc de comparer une demande et des offres de services workflow en considérant l'ontologie et ce conformément aux trois principaux modes de comparaison définis dans (Klush *et al.*, 2006) : le mode Exact, le mode PlugIn et le mode Subsume. Le mode Exact sélectionne une offre si elle correspond exactement à une demande (demande = offre), le mode Plug-In retourne une offre si elle englobe une demande (demande < offre) et le mode Subsume retourne une offre si elle est incluse dans une demande (demande > offre). Les deux derniers modes de comparaison utilisent l'ontologie de domaine. Plus précisément, les offres et demandes de services workflow étant exprimées en OWL-S, nous comparons, selon les trois modes précédents, tous les éléments définis dans les clauses « input » et « output » dans la classe ServiceProfile des offres et des demandes.

L'algorithme de comparaison utilisé à la fois en mode Plug-In et en mode Subsume utilise la fonction Englobe donnée ci-dessous : Englobe (O, D) retourne vrai si l'offre O subsume (englobe) la demande D et faux sinon.

```

Fonction Englobe (E1 : chaîne, E2 : chaîne) : booléen

% Cette fonction retourne vrai si E1 englobe E2 faux sinon
% E1 est un élément de la clause Input ou Output de l'Offre
% E2 est un élément dans la clause Input ou Output
% de la Demande
% A représente l'ontologie (sous forme arborescente)
% On utilise les fonctions de haut niveau suivantes :
% Père(E) : retourne le père de E dans A
% Racine(A) : retourne la racine de A

Variables
  SommetCourant : UnSommet      % Sommet de A en cours d'examen
  LesAncêtres : EnsembledeSommets % Les ancêtres de E2
Début
  LesAncêtres ← ∅
  Si E2 = racine(A) Alors
    % E2 n'a pas d'ancêtre et ne peut pas être englobé
    LesAncêtres ← ∅
  Sinon
    SommetCourant ← Père(E2)
    LesAncêtres ← Père(E2)
    Tant Que (SommetCourant <>Racine(A)) Faire
      SommetCourant ← Père(SommetCourant)
      % «+» désigne l'ajout d'un nouvel élément
      % dans l'ensemble LesAncêtres
      LesAncêtres ← LesAncêtres + SommetCourant
    Fin Tant Que
  Fin Si
  Englobe←(E1 ∈ LesAncêtres)
Fin

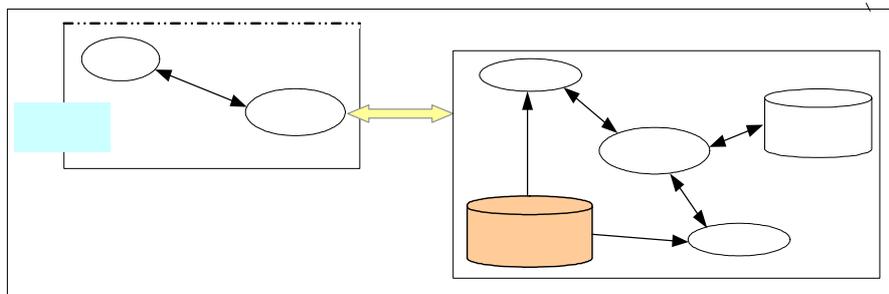
```

## 4. Utilisation de l'Ontologie de Négociation pour la Négociation entre Partenaires

Cette section détaille d'abord l'infrastructure de négociation entre partenaires, présente ensuite l'ontologie de négociation qui regroupe les protocoles de négociation adaptés au contexte du WIO lâche, et montre enfin comment nous utilisons cette ontologie pour choisir dynamiquement un protocole qui servira de support à la négociation.

### 4.1. Infrastructure de Négociation

L'infrastructure de médiation proposée supporte la négociation entre partenaires. (cf. figure 6). Elle se connecte dynamiquement au moteur de workflow via le serveur de connexion et l'interface 6.



**Figure 6.** Infrastructure de médiation pour la Négociation entre Partenaires

Cette infrastructure est organisée autour des composants suivants :

- Un *Serveur de Négociation* qui fournit des informations sur les négociations en cours et notamment les actes de négociation émis par les participants.
- Un ensemble de *Modérateurs* qui gèrent les négociations. Chaque modérateur implémente le protocole sous-jacent à la négociation qu'il gère. Il s'assure que les différents participants impliqués dans la négociation émettent des actes qui respectent les règles du protocole sous-jacent. Un modérateur est dynamiquement créé par le Serveur de Négociation à l'ouverture d'une négociation puis disparaît à la fin de celle-ci. Un modérateur ne gérant qu'une seule négociation, il y a donc autant de modérateurs que de négociations en cours.
- Une base de données *Négociations* qui stocke toutes les informations liées aux négociations en cours telles que l'identité du Serveur de Connexion demandeur, l'identité du Modérateur, la date de création du Modérateur, le protocole de négociation utilisé... Cette base de données est gérée par le Serveur de Négociation.
- une *Ontologie de Négociation* qui stocke les protocoles utilisables au cours d'une négociation. Ce sont ces types de protocoles qui sont en fait implémentables

Agent  
Workflow

Moteur de  
Workflow

Serveur de  
Connexion

par un modérateur, qui peut, par conséquent, être vu comme une instance d'un protocole.

– Un *Sélectionneur* dont le rôle est d'aider au choix d'un protocole de négociation. Le sélectionneur utilise l'ontologie de négociation et est en relation avec le serveur de négociation.

Nous défendons l'idée d'utiliser une ontologie pour décrire les types de protocoles utilisables au cours d'une négociation : cette ontologie permet en effet le partage de ces protocoles qui ne sont ainsi plus codés dans les processus workflow impliqués dans un WIO lâche. Cette ontologie permet ensuite le choix dynamique du protocole de négociation qui va garantir la cohérence sémantique des interactions entre les partenaires impliqués dans un WIO.

#### 4.2. Ontologie de Négociation

Pour définir cette ontologie, nous avons examiné les protocoles suivants, qui sont issus des systèmes multi-agents, et qui sont adaptés à la négociation dans le contexte du WIO lâche : Heuristique, Argumentation, Contract-Net et Enchère, notamment les enchères anglaises, hollandaises, offre cachée meilleur prix et offre cachée deuxième prix encore appelée Vickery (Jennings *et al.*, 2005). Nous avons également utilisé les propriétés suivantes pour caractériser ces protocoles : nombre de partenaires impliqués dans la négociation, comportement des partenaires (compétitif ou coopératif), nombre de tours de négociation (c'est à dire nombre d'interventions de chaque partenaire au cours de la négociation) et nombre d'attributs négociables. Le résultat obtenu est décrit dans la table 1 ci-après.

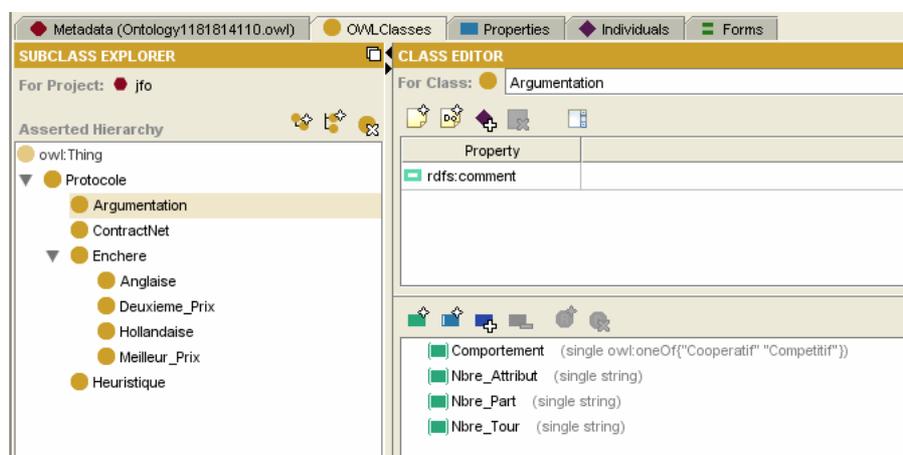
En accord avec cette table, nous avons défini une ontologie de négociation qui spécifie à la fois l'organisation hiérarchique des protocoles, les propriétés et les valeurs correspondantes relatives à ces protocoles. Plus précisément, nous avons défini des classes, des propriétés associées à ces classes et des instances qui représentent les protocoles utilisables pour les négociations dans le WIO lâche. Les propriétés correspondent aux propriétés utilisées pour caractériser les protocoles et les valeurs données à chaque propriété de chaque instance sont extraites de la table de caractérisation des protocoles.

	Comportement	Nombre de Partenaires	Nombre de tours	Nombre d'attributs
Heuristique	compétitif	plusieurs	plusieurs	plusieurs
Argumentation	compétitif	plusieurs	plusieurs	plusieurs
Contract-Net	coopératif	plusieurs	un	plusieurs
Enchère Anglaise	compétitif	plusieurs	plusieurs	un
Enchère Hollandaise	compétitif	plusieurs	plusieurs	un
Enchère Meilleur	compétitif	plusieurs	un	un

Prix				
Enchère	compétitif	plusieurs	un	un
Deuxième Prix				

**Table 1.** *Caractérisation des Protocoles de Négociation*

L'ontologie de négociation présentée ci-dessous a été mise en œuvre avec Protégé-3.2.1 et a donné lieu à la dérivation d'un fichier OWL qui sera utilisé pour la sélection dynamique d'un protocole de négociation. Les figures 7 et 8 décrivent les classes, propriétés et instances de l'ontologie de négociation.



**Figure 7.** *Ontologie de Négociation : Classes et Propriétés*

La figure 7 illustre la hiérarchie des protocoles que nous proposons pour le WIO tâche. Elle montre aussi les propriétés définies pour l'ensemble de ces protocoles. Ces propriétés sont définies dans la classe Protocole et héritées dans les sous-classes de la classe Protocole. La figure 8 présente les instances des classes de l'ontologie : il y a une seule instance par classe. De plus, pour chaque instance nous donnons des valeurs aux propriétés de la classe à laquelle l'instance est rattachée. Par exemple, la figure 8 indique qu'une instance de la classe Heuristique a été définie. Cette instance a pour nom *Protocole\_Heuristique* et les valeurs de ses propriétés indiquent qu'une négociation menée selon ce protocole est de type compétitif (les différents fournisseurs impliqués sont en compétition), que le nombre d'attributs impliqués au cours de la négociation est multiple, que les fournisseurs impliqués sont multiples et que plusieurs actes de négociation sont possibles.

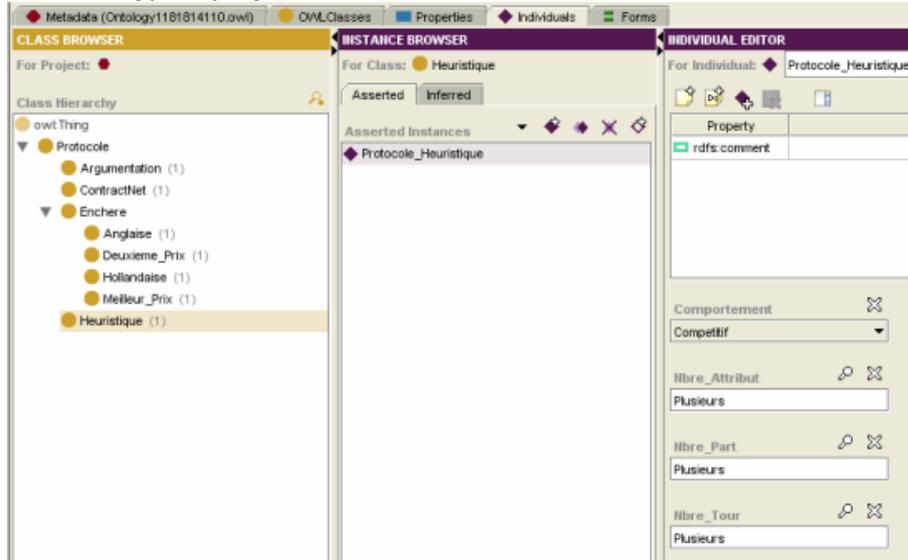
Le code OWL dérivé de cette spécification graphique est partiellement donné ci-dessous à travers quelques éléments participant à la spécification du protocole Heuristique. Ce code sert de point de départ pour exploiter l'ontologie de négociation lors de la sélection dynamique de protocoles.

% Spécification de classe Protocole et de la propriété Nbre\_Part

```

<owl:Class rdf:ID="Protocole"/>
<owl:DatatypeProperty rdf:ID="Nbre_Part">
  <rdf:type rdf:resource="..." />
  <rdfs:domain rdf:resource="#Protocole"/>
  <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
</owl:DatatypeProperty>

```



**Figure 8. Ontologie de Négociation : Instances**

% Spécification de la classe *Heuristique* comme sous-classe de la classe *Protocole*

```

<owl:Class rdf:ID="Heuristique">
  <rdfs:subClassOf rdf:resource="#Protocole"/>
</owl:Class>

```

% Spécification de l'instance *Protocole\_Heuristique* de la classe *Heuristique*

```

<Heuristique rdf:ID="Protocole_Heuristique">
  <Comportement rdf:datatype="...">Competitif</Comportement>
  <Nbre_Attribut rdf:datatype="...">Plusieurs</Nbre_Attribut>
  <Nbre_Part rdf:datatype="...">Plusieurs</Nbre_Part>
  <Nbre_Tour rdf:datatype="...">Plusieurs</Nbre_Tour>
</Heuristique>

```

#### 4.3. Sélection Dynamique d'un Protocole de Négociation

Le principe de fonctionnement du choix du protocole de négociation est le suivant. L'initiateur de la négociation, c'est à dire l'agent workflow de l'organisation demandeur, sollicite (via son serveur de connexion) le serveur de

négociation pour la création d'un modérateur. Le serveur de négociation demande alors à l'agent workflow de caractériser le protocole qu'il souhaite utiliser pour la négociation en donnant des valeurs aux propriétés définies dans l'ontologie. Le serveur de négociation délègue la sélection d'un protocole de négociation à l'agent Sélectionneur qui va rechercher dans le fichier OWL le ou les protocoles qui correspondent aux valeurs définies par l'agent workflow. Ce dernier choisit alors le protocole qu'il souhaite utiliser et transmet son choix au serveur de négociation qui crée le modérateur implémentant le protocole en question. Enfin le serveur de négociation enregistre la création de cette nouvelle négociation dans la base de données Négociations.

L'agent Sélectionneur est sollicité pour cette sélection de protocole de négociation. C'est cet agent qui est capable de sélectionner le ou les protocoles qui répondent aux critères spécifiés par l'agent workflow demandeur en se basant sur les instances définies dans le fichier OWL. L'algorithme de sélection déroulé par cet agent utilise la fonction Chercher\_Protocoles suivante.

```

Fonction Chercher_Protocoles (Crit_Comportement : chaîne,
Crit_Nbre_Attribut : chaîne, Crit_Nbre_Part : chaîne,
Crit_Nbre_Tour : Chaîne) : Ens_Protocoles

% Cette fonction retourne l'ensemble des protocoles
% qui vérifient les différents critères spécifiés par un agent
% workflow demandeur

% TousLesP représente l'ensemble de tous les protocoles
% P représente un protocole de l'ensemble TousLesP
% LesP représente l'ensemble des protocoles sélectionnés

Variables
  P : Protocole
  TousLesP : Ens_Protocoles
  LesP : Ens _Protocoles

Début
  LesP ← ∅
  % On détermine l'ensemble de tous les protocoles
  TousLesP ← Calculer_Ens_Protocoles()

  % On recherche quels protocoles de cet ensemble
  % vérifient les critères
  Pour chaque P dans TousLesP Faire
    % la fonction Comparer détermine pour le protocole P
    % si l'attribut passé en paramètre a pour valeur
    % le critère spécifié par l'agent workflow demandeur
    Si Comparer("Comportement",Crit_Comportement)
      Et Comparer("Nbre_Attribut",Crit_Nbre_Attribut)
      Et Comparer("Nbre_Part",Crit_Nbre_Part)
      Et Comparer("Nbre_Tour",Crit_ Nbre_Tour) Alors
        % «+» désigne l'ajout d'un nouvel élément dans LesP
        LesP ← LesP + P
  
```

```
Fin Si
Fin Pour
Chercher_Protocoles ← LesP
Fin
```

Remarque : on aurait pu utiliser un langage d'interrogation de fichiers OWL tel que nRql (Haarslev *et al.*, 2004) ou OWL-QL (Fikes *et al.*, 2003) pour effectuer cette recherche de protocoles. L'utilisation de tels langages nécessiterait l'introduction de serveur capables d'interpréter ces requêtes (par exemple Racer pour nRql). L'introduction de ces serveurs aurait pour conséquence de compliquer inutilement l'architecture générale de notre modèle de coordination. C'est la raison pour laquelle nous avons spécifié un algorithme supportant la recherche de protocoles.

## 5. Discussion et Conclusion

Cet article a traité de la coordination dynamique de processus workflow impliqués dans un WIO tâche. Plus précisément, cet article a présenté deux infrastructures de médiation définies pour traiter les problèmes de recherche de partenaires et de négociation entre partenaires. Il a également montré comment ces infrastructures utilisent respectivement :

- une *ontologie de domaine* qui aide à connecter une organisation cherchant de l'aide pour réaliser un service workflow à une ou plusieurs organisations capables de mettre en œuvre le service recherché,
- une *ontologie de négociation* qui aide au partage et à la sélection dynamique de protocoles utilisables au cours d'une négociation.

D'autres travaux ont abordé le problème de la coordination dans le WIO tâche (Zeng *et al.*, 2001), (Buhler *et al.*, 2003), (Blake *et al.*, 2003), (Aberg *et al.*, 2005), (Purvis *et al.*, 2005). L'utilisation de l'approche agent est une constante dans ces travaux : les agents sont vus comme un moyen d'assurer la flexibilité des processus workflow impliqués dans le WIO mais également un moyen pour aider à traiter le problème de recherche de partenaires. Les Web services sont également utilisés, sauf dans (Zeng *et al.*, 2001), pour modéliser et implémenter les processus workflow impliqués dans un WIO tâche. La recherche de services est menée à l'aide de registres de découvertes de services (UDDI) mais l'usage d'une ontologie de domaine n'est en aucun cas suggéré dans ces travaux qui, par conséquent, n'exploitent pas, malgré leur approche de type Web, les avantages du Web sémantique et notamment des ontologies, pour traiter des problèmes d'hétérogénéité sémantique que l'on rencontre inévitablement dans le WIO tâche et pour améliorer le mécanisme de filtrage entre une demande et des offres de services.

Par ailleurs, ces travaux se limitent à considérer le problème de recherche de partenaires. Or, trouver des partenaires n'est pas suffisant pour choisir le meilleur : une étape de négociation, en terme de prix, de délai ou encore de manière

d'implémenter le service, est nécessaire pour évaluer chaque partenaire. Définir une ontologie pour les protocoles de négociation simplifie la conception d'un processus de WIO lâche : il y a partage du code relatif à la négociation qui n'est, par conséquent, plus encapsulé dans les processus impliqués dans un WIO lâche. Utiliser une ontologie pour les protocoles de négociation, est selon nous, une première étape vers la négociation automatique dans ce domaine puisque l'on permet la sélection dynamique de protocoles. Une telle approche a été adoptée dans le contexte du commerce électronique (Tamma *et al.*, 2005), mais pas, à notre connaissance, dans le domaine du WIO lâche.

Nous avons développé MatchFlow qui est un prototype validant notre approche pour la recherche de partenaires (Bouzuenda, 2006). MatchFlow est développé à l'aide de l'outil MadKid ([www.madkit.org](http://www.madkit.org)) qui supporte java et fonctionne en mode distribué. Plus précisément, MatchFlow implémente un matchmaker, des agents workflow fournisseur, un agent workflow demandeur et leurs serveurs de connexion associés. Comme indiqué précédemment, et défendu dans (Andonoff *et al.*, 2005), les offres et demandes de services workflow sont spécifiées avec le formalisme des Réseaux de Pétri à Objets (Sibertin, 1985) et publiées avec OWL-S. Nous avons mis en œuvre l'étude de cas «Relecture des articles soumis à une conférence» qui relève du WIO lâche et avons implémenté l'ontologie de domaine «Conference ». Nous avons également évalué les performances du matchmaker à la fois en termes de *temps de réponse* et de *qualité de réponse* (c'est à dire qu'est ce que le matchmaker retourne comme offre pour une demande donnée) lors des opérations de comparaison en considérant les trois modes présentés précédemment. Le lecteur intéressé peut consulter (Bouzuenda, 2006) pour de plus amples informations sur l'étude de cas et l'évaluation des performances du matchmaker.

Nos travaux actuels portent sur la mise en œuvre de l'infrastructure de négociation entre partenaires dans le cadre du prototype NegoFlow (cf. Annexe 1). Plus précisément, nous avons pour l'instant mis en place l'ontologie de négociation ainsi que quelques agents définis dans l'infrastructure de négociation et dans le moteur de workflow à savoir le serveur de négociation, le serveur de connexion, l'agent workflow demandeur, les agents workflow fournisseurs et le sélectionneur (cf. Annexe 2). Il nous faut maintenant développer les modérateurs implémentant les différents types de négociation que nous proposons pour le WIO lâche.

## **Bibliographie**

- Aberg C., Lambrix C., Shahmehri N., «An Agent-Based Framework for Integrating Workflows and Web Services», *14th International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE'05*, Linköping, Sweden, June 2005, p. 27-32.
- Andonoff E., Bouzuenda L., Hanachi C., «La Coordination dans le Workflow Inter Organisationnel Lâche : une Approche basée sur les Agents et le Web sémantique». *Revue Ingénierie des Systèmes d'Information*, Vol. 11, n°4, 2006.

- Andonoff E., Bouzguenda L., Hanachi C., «Specifying Workflow Web Services for Finding Partners in the context of Loose Inter-Organizational Workflow». *Proceedings of the 3rd International Conference on Business Process Management, BPM'05*, Nancy, France, September 2005, p. 120–136.
- Blake B., Goma H., «Agent-oriented Compositional Approaches to Services-based Cross Organisational Workflow». *Journal on Decision Support Systems*, June 2004, p. 176–181.
- Bouzguenda L., «Coordination Multi-Agents pour le Workflow Inter Organisationnel Lâche». Thèse de Doctorat, Université de Toulouse 1, mai 2006.
- Buhler P., Vidal J., «Towards Adaptive Workflow Enactment using Multiagent Systems». *Journal on Information Technology and Management*, Vol. 6, 2005, p. 61–87.
- Divitini M., Hanachi C., Sibertin-Blanc C., «Inter-Organizational Workflow for Enterprise Coordination. Coordination of Internet Agents», chapter 15, A. Omicini, F. Zambonelli, M. Klusch, R. Tolksdorf Editors, Springer Verlag, 2001.
- Fikes R., Hayes P., Horrocks I., OWL-QL - A Language for Deductive Query Answering on the Semantic Web. Knowledge Systems Laboratory, Stanford University, Stanford, California, USA, 2003.
- Haarslev V., Moeller R., Wessel M., «Querying the Semantic Web with Racer and nRQL». *3rd International Workshop on Applications of Description Logic, ADL'04*, Ulm, Germany, 2004.
- Hernandez N., «Ontologies de domaine pour la modélisation du contexte en recherche d'information». Thèse de Doctorat, Université Toulouse 3, 2005.
- Jenning N., Faratin P., Lomuscio A., Parsons S., Sierra C., Wooldridge M., «Automated Negotiation: Prospects, Methods and Challenges». *Journal on Group Decision and Negotiation*, Vol. 10, n° 2, 2001, p. 199–215.
- Klusch M., Fries B., Sycara K., «Automated Semantic Web Service Discovery with OWLS-MX». *Proceedings of the 5th International Conference on Autonomous Agents and Multi-Agents Systems, AAMAS'06*, Hakodate, Japan, 2006, p. 915–922.
- Medjahed B., Bouguettaya A., Elmagarmid A., «Composing Web Services on the Semantic Web». *Journal on Very Large DataBases*, Vol. 12, n°4, 2003, p. 333–351.
- Lando P., «Conception et développement d'applications informatiques utilisant des ontologies : application aux EIAH». *Actes des 1ères Rencontre des Jeunes Chercheurs en EIAH, RJC-EIAH'06*, Evry, France, Mai 2006.
- Protégé-2000, Informations accessibles à <http://protege.stanford.edu/>
- Savarimuthu T., Purvis M., Purvis MK., Cranefield S., «Integrating Web Services with Agent Based Workflow Management System», *Proceedings of the 5th International Conference on Web Intelligence, WI'05*, Compiègne, France, 2005, p. 471–474
- Sibertin-Blanc C., «High Level Petri Nets with Data Structure». *Proceedings of the 6th International Workshop on Petri Nets and Applications*, Espoo, Finland, 1985.

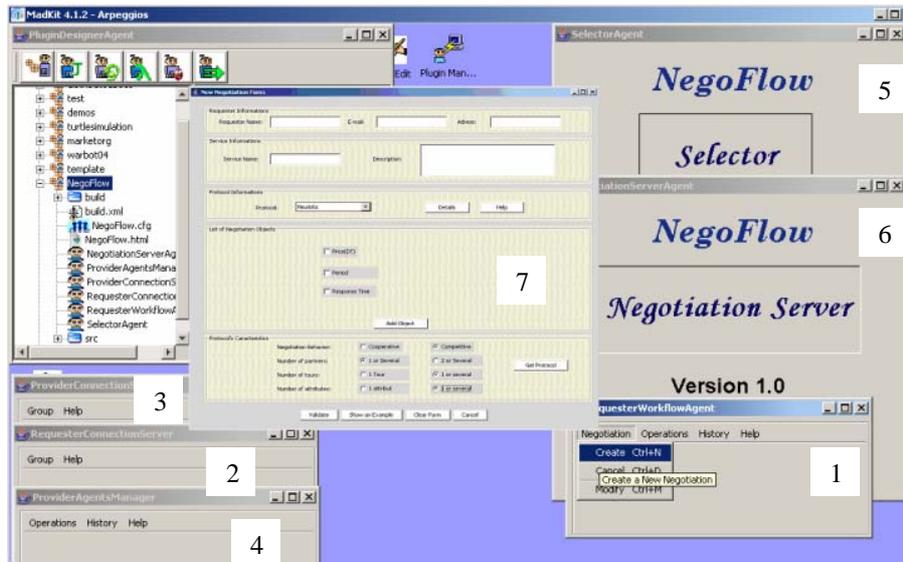
Tamma V., Phelps S., Dickinson I., Wooldridge M., «Ontologies for Supporting Negotiation in E-Commerce». *Engineering Applications of Artificial Intelligence*, Vol. 18, n°2, 2005, p. 223–236.

Workflow Management Coalition, «The Workflow Reference Model». Technical Report WFMC-TC-1003, Workflow Management Coalition, November 1994.

Zeng L., Ngu A., Benatallah B., O'Dell M., «An Agent-Based Approach for Supporting Cross-Enterprise Workflows». *Proceedings of the 12th Australian Database Conference*, ADC'01, Bond, Australia, February 2001, p. 123–130.

## Annexe 1 : Aperçu de NegoFlow

Cette annexe présente le prototype NegoFlow. Plus précisément, elle montre un ensemble d'agents se coordonnant lors de la phase de négociation entre partenaires : un agent demandeur (1), son serveur de connexion (2), un agent fournisseur, son serveur de connexion (3) et son agent manager (4), l'agent Serveur de Négociation (5) et l'agent Sélectionneur (6).



La fenêtre 7 montre comment, lors de la création d'une négociation, l'agent Sélectionneur aide au choix du protocole de négociation sous-jacent. Cette fenêtre est détaillée en annexe 2 ci-dessous.

## Annexe 2 : Spécification des critères pour le choix du protocole

Cette annexe montre comment l'agent Sélectionneur aide au choix du protocole. Cet agent permet tout d'abord de sélectionner dans une série d'options les

caractéristiques du protocole qui sera sous-jacent à la négociation : nombre de tours, nombre d'attributs, nombre de partenaires et attitude des partenaires lors de la négociation (cf. ovale rouge). Il retourne ensuite le ou les protocoles qui vérifient ces critères. L'agent workflow demandeur en choisit alors un et transmet son choix, via son serveur de connexion, au serveur de négociation qui crée en conséquent une négociation implémentant le protocole en question.

**New Negotiation Form**

**Requester Informations**  
Requester Name:  E-mail:  Address:

**Service Informations**  
Service Name:  Description:

**Protocol Informations**  
Protocol:

**List of Negotiation Objects**  
 Price(DT)  
 Period  
 Response Time

**Protocol's Characteristics**  
Negotiation Behavior:  Cooperative  Competitive  
Number of partners:  1 or Several  2 or Several  
Number of tours:  1 Tour  1 or several  
Number of attributes:  1 attribut  1 or several