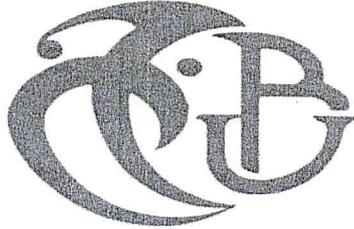


République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

**Université Saad Dahlab Blida**

N° D'ordre : .....



Faculté des sciences

**Département d'informatique**

Mémoire Présenté par :

Taieb Errahmani asma      Bouzidi nawel

**En vue d'obtenir le diplôme de master**

Domaine : Mathématique et informatique

Filière : Informatique  
Spécialité : Informatique  
Option : Ingénierie de logiciel

Sujet :

*Annotation sémantique des Workflow inter-organisationnels basée sur le framework "Open Annotation"*

Soutenu le :

M. Boustia	Président
M. Hammouda	Examineur
M. Rezzeq	Examineur
M. Abbassen Ali	Promoteur
Mme. Chicki Imane	Encadrante

MA-004-262-1

**Promotion**  
2014 / 2015

## Résumé

Dans ce mémoire, nous proposons une solution au problème d'incompatibilité sémantique des workflows, à travers l'enrichissement de leur description syntaxique par une couche sémantique, qui explicite le sens des données. Nous proposons donc, un procédé d'annotation sémantique, qui associe un concept tiré à partir d'une ontologie métier adéquate à un élément syntaxique d'un workflow. Le mécanisme d'annotation utilisé est inspiré de celui proposé par le framework "Open Annotation".

L'objectif de l'annotation sémantique des workflows inter-organisationnels est de permettre la résolution des problèmes d'interopérabilité et d'incompatibilité entre les différents partenaires de la collaboration. Ces incompatibilités surgissent lorsque la sémantique des données échangées entre les workflows partenaires est différente. D'un point de vue conceptuel, les éditeurs (designers) de workflows actuels, ne prennent en considération que la description syntaxique des workflows, selon différents langages et formats comme BPEL, YAWL, BPMN ou XPDL.

Open Annotation, n'étant pas utilisé pour la description sémantique des workflows, mais beaucoup plus pour l'annotation des données multimédia sur le Web, n'empêche qu'il présente divers avantages par rapport aux procédés d'annotation sémantique classiques, tel que la possibilité de spécifier la portion exact du workflow à annoter, la spécification des raisons exactes de l'annotation et le partage et la réutilisation des annotations.

Nous avons utilisé en final un éditeur de workflows open-source, qui est Oryx editor, pour y incorporer notamment un plugin qui gère les annotations sémantique selon le format "Open Annotation". Il a aussi été question d'étendre l'ontologie "Open Annotation", pour y inclure les éléments spécifiques au langage de description de workflows XPDL (XML Process Definition Language) où nous proposons l'ontologie spécialisée OA4XPDL (Open Annotation for XPDL).

## ملخص

الهدف من الشرح الدلالي لألية سير العمل داخل المؤسسات هو إيجاد حلول لمختلف مشاكل عدم التوافق بين الشركاء. يظهر هذا النوع من المشاكل عادة إذا كانت دلالات البيانات المتبادلة بين الشركاء مختلفة. من وجهة نظر مفاهيمية جميع البرامج الحالية لتصميم وإنشاء آليات سير العمل داخل المؤسسات تأخذ بعين الاعتبار الوصف النحوي فقط و تستعمل عدة لغات و الأشكال في هذا العمل، لذلك اقترحنا حل لمشكل عدم التوافق الدلالي بين مختلف آليات سير العمل و هذا من خلال إثراء وصفهم النحوي بأجزاء الوصف الدلالي الذي يتضمن معاني المعطيات، فكان اقتراحنا لهذه المنهجية التي تحتوي على مجموعة مفاهيم مستنبطة من انطولوجيا مرتبطة هي الأخرى بعناصر نحوية لهاته الآليات. هاته المنهجية المستعملة في مشروعنا تستند على مفهوم " الوصف الدلالي المفتوح " و هذا الأخير لم يستعمل سابقا في شرح و تدقيق آليات سير العمل داخل المؤسسات بل كان مقتصر على مجال الويب. في هذا المشروع إستعملنا كبرنامج سير العمل المفتوح المصدر، المحرر " اوريكس "، لدمج البرنامج المساعد الذي يعالج الشروحات الدلالية وفقا لصيغة " الوصف الدلالي المفتوح ". و أيضا مددنا الأنطولوجيا الخاصة بها لتشمل عناصر محددة جديدة أضيفت أثناء إنجازنا لهذا العمل و لذلك كانت النتيجة تشكيل انطولوجيا جديدة مكملة للأنطولوجيا الأولى المستعملة.

## Abstract

The objective of semantic annotation inter-organizational workflows is to resolve interoperability problems and incompatibility between different collaboration partners. These incompatibilities arise when the semantics of the exchanged data between the inter-organizational workflows is different. From a conceptual point of view, the actual editors (designers) of workflows only consider the syntactic description, with different languages and formats such as BPEL, YAWL, BPMN and XPD L.

In this work, we propose a solution to the problem of semantic incompatibility of workflows, through the enrichment of their syntactic description by a semantic layer, which explains perfectly the data meaning. So we propose a method for semantic annotation, which combines a concept drawn from a proper business ontology to a syntax element of a workflow. The annotation mechanism used is based on the one proposed by the "Open Annotation" framework.

Open Annotation, was not used for the semantic description of workflows, but much more for the annotation of multimedia data on the Web. Still it has various advantages over the classical semantic annotation processes. such as the ability to specify the exact portion of the workflow to annotate, specifying exact reasons of the annotation and sharing and reuse annotations.

So in our project we used an open-source workflow editor, which is Oryx editor, to incorporate a plugin that handles semantic annotations according to the "Open Annotation" format. It was also a question of extending the ontology "Open Annotation", to include specific elements to the description language of workflow XPD L (XML Process Definition Language) where we offer the special ontology OA4XPDL (Open Annotation for XPD L).

# REMERCIEMENTS

*En terminant notre mémoire de fin d'études et avant de présenter notre travail, il est de notre devoir d'exprimer nos remerciements tout d'abord à dieu tout puissant qui nous a donné la force, la patience et la volonté d'achever ce mémoire.*

*Puis à nos parents qui nous ont été d'un grand soutien pendant toute la période de nos études et dès le début de ce travail.*

*Nous tenons à remercier Monsieur le directeur du Centre de Développement des Technologies Avancée (CDTA) d'Alger; pour nous avoir accepté au sein du Centre et pour l'expérience enrichissante et pleine d'intérêt qu'il nous a fait vivre durant notre stage.*

*Nos remerciements à tous les enseignants de la faculté de des sciences de BLIDA et surtout nos enseignants du département informatique.*

*Nous tenons à exprimer nos profondes gratitudes et nos vifs remerciements à notre promoteur M. Ali Abbassene responsable de l'équipe Collaboration Inter-Organisationnelle basée Web du CDTA, pour nous avoir accueilli au sein du Centre, pour le sujet qu'il nous a proposé et pour nous avoir accordé toute sa confiance ; pour le temps qu'il nous a consacré tout au long de cette période, sachant répondre à toutes nos interrogations ; sans oublier sa participation au cheminement de ce travail.*

*Nous vous souhaite remercier notre encadrant Mme Chicki pour son aide précieuse à la rédaction de ce mémoire, sa patience, ainsi que pour ses conseils et sa disponibilité.*

*Nos remerciement aux membres du jury pour nous avoir fait l'honneur de d'examiner et d'apprécier notre travail.*

*Nous remercions, de tout cœur, tous ce qui ont contribué de près ou de loin la réalisation de ce travail.*

*A tous...*

*Merci Beaucoup*

## DEDICACE

*C'est avec une immense fierté que je dédie ce modeste travail de fin d'études à ceux qui ont fait de moi ce que je suis aujourd'hui, grâce à leurs sacrifices, leurs tendresses et leur amour : mes très chère parents que dieu me les garde*

*A toute ma petite famille : mes frères Youcef, Walid et sa femme , ma sœur Meriem et son époux, pour leur soutien moral et leurs encouragements, un grand merci à eux*

*Mes anges : Mehdi et Amir.*

*Ma petite princesse :Basma*

*Mes grands-parents, ma tante Chafika, et tous mes oncles.*

*A toute mes cousines .*

*A Mes amies : Fatiha,hadjer,Mounia,Naima et Khadija*

*Mon très cher binôme : Nawel et toute sa famille.*

*A notre sœur Khadija Ameur pour leur aide et encouragements.*

*A notre promoteur Ms Ali Abbassene pour son aide précieuse et pour le temps qu'il nous a consacré.*

*A toutes les personnes qui nous ont Prodigué des encouragements et se sont donné la peine de nous soutenir durant cette formation.*

*Asma TAIEB ERRAHMANI*

## DEDICACE

*C'est avec une immense fierté que je dédie ce modeste travail de fin d'études à ceux qui ont fait de moi ce que je suis aujourd'hui, grâce à leurs sacrifices, leurs tendresses et leur amour : mes très chères parents que Dieu me les garde*

*A toute ma petite famille : ma sœur Naima, mes frères Abd el Hakim et son fiancée Mounia, Abd el Wahab et son fiancée Nadjat, à mon fiancé Hamza pour leur soutien moral et leurs encouragements, un grand merci à eux*

*tous mes oncles et mes tantes.*

*A toute mes cousines .*

*A toute la famille Medjber.*

*A Mes amies : Soumia , Sabrina , Kholoud ,Hadjer , Khadija , Naima*

*Mon très cher binôme : asma et toute sa famille.*

*A notre sœur Khadija Ameur pour leur aide et encouragements.*

*A notre promoteur Ms Ali Abbassene pour son aide précieuse et pour le temps qu'il nous a consacré.*

*A toutes les personnes qui nous ont prodigué des encouragements et se sont donné la peine de nous soutenir durant cette formation.*

*Nawel Bouzidi*

## Table des Matières

<b>Introduction Générale</b> .....	1
<b>Chapitre 01: Workflow</b> .....	3
1.1 Introduction.....	4
1.2 Gestion des processus métiers ( Business Process Management ).....	4
1.2.1. Processus Métier.....	4
1.2.2. Gestion des Processus Métiers (BPM).....	4
1.3 Workflow.....	5
1.3.1. Définitions.....	5
1.3.2. Concept de base .....	5
1.3.3. Système de gestion de Workflow (SGWF).....	6
1.3.4. Workflow inter- organisationnel .....	8
1.3.5. Langages de modélisation de workflow.....	9
1.3.6. Langages d'exécution de workflow.....	12
1.3.7. Editeurs Workflows .....	13
1.4 Conclusion.....	14
<b>Chapitre 02 : Description sémantique des Workflows</b> .....	15
2.1 Introduction.....	16
2.2 Les services Web.....	16
2.2.1 Web Service Description Language(WSDL).....	17
2.2.2 Simple Object Access Protocol (SOAP).....	17
2.2.3 Universal Description, Discovery and Integration (UDDI).....	18
2.3 La sémantique dans les services Web et Workflow.....	18
2.4 L'annotation sémantique .....	18
2.5 Annotation sémantique de langages de Workflows .....	19

2.5.1 Semantic Annotation for WSDL ( SAWSDL) .....	20
2.5.2 Semantic Service Markup ( SESMA ) .....	20
2.5.3 Open annotation .....	21
2.6 Conclusion.....	30
<b>Chapitre 03:Ontologie.....</b>	<b>31</b>
3.1 Introduction .....	32
3.2 Les Ontologies.....	32
3.2.1 Définitions.....	32
3.2.2 Composants d'une ontologie.....	32
3.2.3 Langages de représentation des ontologies.....	34
3.2.4 Les éditeurs d'ontologie.....	36
3.3 SCOR (Supply Chain Operation Reference) .....	36
3.4 Conclusion.....	37
<b>Chapitre 04 :Les Choix techniques et méthodologiques .....</b>	<b>38</b>
4.1 Introduction.....	39
4.2 Choix de l'outil approprié.....	39
4.2.1 Critères de choix .....	39
4.2.2 logiciels.....	40
4.2.3 Le choix final.....	42
4.3 Langages de modélisations et d'exécution utilisés.....	42
4.3.1. Langage de modélisation choisi.....	42
4.3.2. Langage d'exécution choisi.....	43
4.4 l'approche utilisé pour la sémantique des Workflow.....	43
4.5 Les ontologies utilisées.....	43

4.6 Conclusion.....	43
<b>Chapitre 05: Description et mise en œuvre de la solution proposée.....</b>	<b>44</b>
5.1 Introduction.....	45
5.2 Limite des technologies Workflow.....	45
5.3 Description du travail réalisé .....	45
5.4 Modélisation des nouvelles fonctionnalités fournies par notre procédé.....	46
5.5 Eléments du Workflow à annoter.....	49
5.6 Architecture et fonctionnement de plugin ORYX .....	50
5.7 Structuration du projet Oryx.....	50
5.8 Les langages et l'environnement de développement .....	51
5.8.1 Les langages de programmation utilisées.....	51
5.8.2 Environnement de développement .....	52
5.9 Intégration de l'annotation sémantique dans oryx editor .....	52
5.10 La réalisation de la solution.....	54
5.11 La démarche d'une annotation sémantique d'un Workflow inter-organisationnel dans oryx editor personnalisé.....	59
5.12 Conclusion.....	64
<b>Conclusion Générale.....</b>	<b>65</b>
<b>Liste des Abréviations.....</b>	<b>66</b>
<b>Bibliographie.....</b>	<b>67</b>
<b>Annexe</b>	
Annexe A : CONFIGURATION DE L'ENVIRONNEMENT DE DÉVELOPPEMENT...1	
Annexe B : BPMN1.2.....	4
Annexe C : XPDL 2.1.....	8
Annexe D : Ontologie OA4XPDL.....	13

# La liste des figures

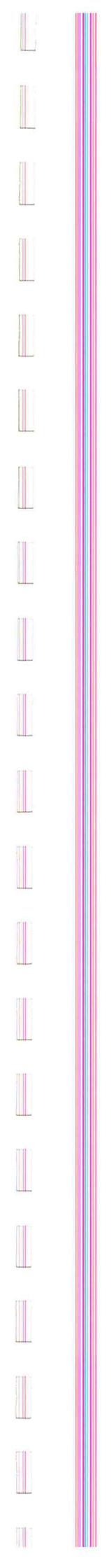
Figure 1.1: Le modèle de référence du workflow .....	7
Figure 1.2 : Exemple d'un Workflow en EPC.....	10
Figure 1.3: Diagramme d'activité en UML.....	11
Figure 2.1 Fonctionnement des services Web.....	17
Figure 2.2. Annotation linguistique d'un document.....	19
Figure 2.3: Description de l'opération devis de l'assuré en SAWSDL.....	20
Figure 2.4: Description SESMA pour l'annotation sémantique d'une opération WSDL.....	21
Figure 2.5: Les concepts « Annotation », « Body » et « Target » du modèle open annotation	22
Figure 2.6: Modèle de base d'Open Annotation.....	23
Figure 2.7: Typage d'un Body et Target. ....	23
Figure 2.8: Exemple d'une motivation.....	25
Figure 2.9: Specifieurs et Specific Ressource.....	25
Figure 2.10: Selector.....	26
Figure 2.3.Fragment Selector. ....	28
Figure 3.4: Exemple d'un concept.....	33
Figure 3.5: La pyramide des langages basés Web.....	34
Figure 3.3: les processus de gestion de base de modèle de référence de SCOR opération.....	37
Figure 4.6. Un aperçu de la pile de l'outil Activiti.....	40
Figure 5.1: Architecture générale d'Oryx.....	46
Figure 5.2: Les fonctionnalités ajoutées dans oryx editor .....	46
Figure 5.3: La démarche utilisée pour annoter sémantiquement les Workflows inter-organisationnels.....	47
Figure 5.4: Les composants du Projet d'Oryx .....	50

Figure 5.5: La démarche utilisée pour annoter sémantiquement les Workflows inter-organisationnels.....	51
Figure 5.6: Structure du code source d'Oryx.....	54
Figure 5.7: Un Aperçu originale de oryx editor.....	56
Figure 5.8: Le stencil set "OA-BPMN1.2" .....	56
Figure 5.9: Les éléments du Stencil set.....	57
Figure 5.10: Les propriétés associer au Annotation.....	58
Figure 5.11: Les propriétés associer au Body.....	58
Figure 5.12: Les propriétés associer au Target.....	58
Figure 5.13: Les rôles associer au Annotation.....	58
Figure 5.14: Le nouveau plugin d'exportation.....	59
Figure 5.15: L'authentification .....	59
Figure 5.16: Vérification de l'authentification.....	60
Figure 5.17: Résultat de l'authentification.....	60
Figure 5.18: Workflow inter-organisationnel.....	61
Figure 5.19: Annotation sémantique d'une activité d'un workflow inter-organisationnel.....	63
Figure 5.20: Annotation sémantique de deux activités d'un workflow inter-organisationnel.....	63
Figure 5.21: Le fichier résultat via l'exportation.....	64

## La liste des tableaux

Tab 2.1: Tableau représente les namespaces définis par OA .....	22
Tab 2.2. Tableau représente l'ensemble des instances de la classe OA .....	25
Tab 2.3. Tableau représente une description sur la classe et la relation du selector. ....	26

Tab 2.4. Tableau représente un model de fragment selector. ....	27
Tab 4.1. Tableau de critères de sélection de logiciels libres.....	39
Tab 4.2. Tableau comparaison entre les différents logiciel de workflow.....	42



# *Introduction Générale*

## Introduction générale

Aujourd'hui, La collaboration interentreprises constitue un facteur de différenciation décisif dans l'environnement professionnel international. C'est grâce à la collaboration, qu'on peut résoudre des problèmes de façon collective, faire face à un marché très concurrentiel, et créer de la valeur pour les clients. En outre, les entreprises peuvent obtenir d'importants avantages en termes de performances pour accélérer la productivité et la bonne prise de décision.

Le B2B (business-to-business) est un outil important ayant pour objectif de réaliser la collaboration entre entreprises. Toutefois, pour qu'une entreprise puisse pleinement en profiter, il lui faudra pouvoir aisément aligner son système d'information d'une part avec les métiers de l'entreprise elle-même et d'une autre part avec les métiers des entreprises avec lesquelles elle va collaborer. Aussi, les technologies du Business Process Management (BPM) ont impulsé de nouvelles approches qui permettent de mettre en place des architectures d'entreprise plus souples, plus évolutives, et plus aptes à satisfaire les besoins d'agilité.

L'échange d'informations et de services entre les entreprises collaboratrices se réalise à travers les communications qui relient leurs workflows métiers respectifs. Dans un workflow métier, les activités offertes vers l'extérieur sont souvent implémentées sous forme de service Web. La mise en relation des workflows partenaires nécessite de mettre en correspondance les activités en relation, ainsi est réalisé l'interopérabilité des workflows interentreprises.

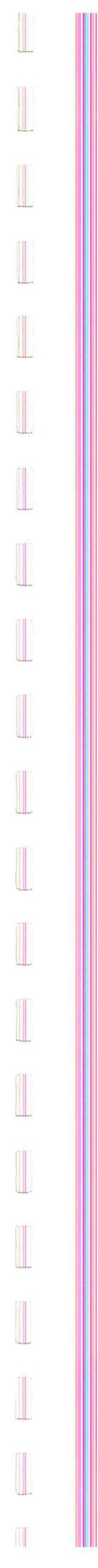
L'interopérabilités interentreprises nécessite tout d'abord une bonne intercompréhension entre les différents workflows y afférents. En effet, la description syntaxique des workflows ne permet pas seule de gérer la complexité des échanges entre les partenaires, ce qui rend impératif l'ajout d'une couche de sémantique afin de régler les problèmes d'hétérogénéité et de rendre les échanges entre les workflows plus cohérents.

Notre travail va s'orienter dans ce sens, où l'on va tenter de mettre en place un outil de modélisation et de description sémantique des workflows inter-organisationnels, qui donne des solutions potentielles aux problèmes d'hétérogénéité entre workflows collaboratifs et exempt de toute incompatibilité sémantique en mettant à profit le nouveau mécanisme d'annotation « Open Annotation ».

Nous avons ainsi organisé ce mémoire en deux parties. La première traite de l'état de l'art et de la définition de la problématique (Chapitres 1, 2, 3 et 4), elle s'intéresse à l'étude du contexte de notre travail où on fera un bref tour d'horizon sur les différents aspects liés à la communication entre workflows. On y présentera aussi la problématique qui a motivé notre contribution. La seconde partie du mémoire couvre le dernier chapitre et concerne la contribution en elle-même. Et par la suite elle présente la mise en œuvre et l'implémentation des différents modules développés.

- **Chapitre 1** : Intitulé « **Workflow** », dans ce chapitre on donnera une définition des workflows, leur concepts, et leur extension en workflows inter-organisationnels, ainsi que les différents langages de modélisation et d'exécution et les différents éditeurs existants.
- **Chapitre 2**: Intitulé « **Description sémantique des Workflows** », dans ce chapitre on se penchera sur l'intérêt d'associer une description sémantique à un workflow, ainsi que les différents mécanismes existants pour réaliser cette annotation.
- **Chapitre 3**: Intitulé « **Ontologies** », on va présenter dans ce chapitre ce concept important pour la réalisation de l'annotation sémantique d'un workflow inter-organisationnel.
- **Chapitre 4**: Intitulé « **Choix techniques et méthodologiques** », ce chapitre résume toute nos choix concernant les langages, les logiciels et des techniques, pour la réalisation de notre projet.
- **Chapitre 5**: Intitulé « **Description et mise en œuvre de la solution proposée** », dans ce chapitre on présente la modélisation de la solution proposée; par la suite nous allons présenter les résultats concrets réalisés.

Enfin, ce mémoire se termine par une conclusion générale qui présente un bilan du travail réalisé et des perspectives.



# *Chapitre 01*

## *Workflow*

### I. Introduction

La bonne compréhension du fonctionnement interne de l'entreprise mène à une communication efficace entre les différents acteurs qui la constituent. C'est pourquoi les entreprises ont structuré leur organisation autour de **processus métiers** décrivant leurs fonctionnements. Cela leur permet d'avoir une vision globale et précise de leur métier et ainsi d'améliorer et optimiser leurs processus métiers.

Dans ce chapitre, nous allons présenter les différentes notions et définitions des éléments et des outils nécessaires pour avoir une bonne modélisation d'une entreprise.

### II. Gestion des processus métiers ( Business Process Management )

#### II.1. Processus Métier

Dans une entreprise, un produit ou un service est en général le résultat d'un enchaînement d'actions ayant un déclencheur commun reliées entre elles par des flux d'information, selon des règles métier, réalisées par différents acteurs bien identifiés. L'enchaînement de ces actions est appelé un processus métier. (Soussi, 2013a)

Il consiste aussi en un ensemble d'activités exécutées et coordonnées dans le cadre d'un environnement organisationnel et technique, pour réaliser un objectif prédéterminé. Un processus métier est généralement affecté à une seule organisation, mais peut interagir avec d'autres processus métiers appartenant à d'autres organisations dites partenaires. (Baouab, 2013a)

Selon la Workflow Management Coalition (WfMC)<sup>1</sup> un processus métier est défini comme : *« un ensemble de procédures et d'activités plus ou moins liées qui réalisent collectivement un objectif métier, en général au sein d'une structure organisationnelle définissant des rôles et des relations fonctionnelles. Un processus métier peut être entièrement inclus dans une organisation simple ou peut s'étendre sur plusieurs organisations. »*. (Souissi, 2013b)

#### II.2. Gestion des Processus Métiers (BPM)

La gestion des processus métiers est la discipline qui fournit l'ensemble des méthodes, technologies et outils destinés à améliorer l'efficacité, la traçabilité et l'agilité des processus métiers au sein desquels collaborent des systèmes, des logiciels, des personnes et aussi des clients, des fournisseurs et des partenaires. (Xavier, 2009)

Les systèmes d'information jouent un rôle important dans la gestion des processus métiers, vu que la plupart des entreprises de nos jours ont tendance à automatiser leurs processus et que leurs activités sont de plus en plus supportées par les systèmes d'information. En effet, les activités d'un processus métier peuvent être exécutées manuellement, par des employés ou à l'aide du système d'information, dans le but de réduire les temps de production et de simplifier l'interaction des différents acteurs. Cela permet aux acteurs de se concentrer sur la valeur ajoutée. (Souissi, 2014c)

<sup>1</sup> WfMC: Organisation internationale de normalisation dans le domaine du Workflow ( <http://www.wfmc.org/>).

Et dans le même contexte, les experts métier d'une organisation gèrent particulièrement leurs processus métier, se portant principalement sur la représentation du processus dans une forme qui supporte l'automatisation comme la modélisation, jusqu'à l'exécution interactive avec les acteurs et machines impliquées dans le processus. Dans ce cas, on parle de Workflow qui présente une vision automatisable d'un processus métier. (Souissi, 2014d)

### III. Workflow

Aujourd'hui les entreprises gèrent un grand nombre d'informations et doivent les coordonner, ce qui nécessite une certaine flexibilité et une adaptation rapide au changement du marché. Une nouvelle technologie a été proposée pour répondre d'une manière efficace à ces besoins, qui est la technologie Workflow.

L'évolution de cette technologie a couvert un certain nombre de produits différents ce qui soulève la nécessité de la standardisation. La WfMC a été fondée en 1993, comme une organisation à but non lucratif mais ayant pour objectif de soutenir l'utilisation de la technologie Workflow, grâce à l'élaboration de normes d'interopérabilité et de connectivité. Aujourd'hui la WfMC possède environ 300 membres représentant des développeurs, des analystes et des groupes de recherche. (Jonuzi, 2014a)

#### III.1. Définitions

Selon la WfMC, la technologie Workflow est définie comme « *L'automatisation d'un processus d'affaires, en tout ou partie, au cours de laquelle les documents, informations ou les tâches sont passés d'un participant à l'autre pour l'action, selon un ensemble de procédure et règles.* ». (Jonuzi, 2014b)

La technologie Workflow est aussi définie comme étant « *La modélisation et la gestion informatisée de l'ensemble des tâches à accomplir et des différents acteurs appliqués dans la réalisation d'un processus métier d'une organisation.* ». (Andre & Tchamga, 2010)

Elle est aussi défini comme « *Les aspects opérationnels d'un processus : la séquence des tâches et qui les réalisent, le flot de données qui supporte ces tâches, et les mécanismes qui permettent de mesurer, suivre et contrôler ces tâches.* ». (Mohan, 1999)

#### III.2. Concepts de base

Les concepts de base de la technologie Workflow sont expliqués par la métaphore des "3R" (Routes, Rules, Rôles) : (Sini, 2013)

##### ➤ Le routage :

Le routage des documents, des informations, ou des tâches a été la première grande fonction de la technologie Workflow. C'est à dire les chemins que prennent les différents résultats d'une activité à l'autre, d'un rôle à l'autre et donc, d'un participant à l'autre. L'optimisation d'un Workflow est réalisée lorsqu'on spécifie l'ordonnancement des interdépendances fondamentales mais qu'on laisse quand même une petite marge de manœuvre à l'acteur afin de rester flexible.

➤ **Les règles :**

La gestion des règles de coordination des activités est complémentaire au routage, car l'itinéraire d'un processus dépend des règles qui définissent à la fois la nature des informations ainsi que leur modalité de transport d'une personne à l'autre. Et donc elles sont indispensables au Workflow.

➤ **Les rôles :**

Finally, il faut gérer les différents rôles, le troisième "R" de la métaphore. Il s'agit de gérer les différentes personnes, qui accomplissent les tâches et qui communiquent entre elles. Ici il est important de noter qu'en fait, on ne gère pas des personnes en tant qu'individus mais en tant que rôles, c'est à dire des fonctions. Ces rôles ne correspondent pas nécessairement à des personnes, car les tâches ne sont pas nécessairement réalisées par des personnes. Router un courrier électronique, faire des calculs complexes sont des exemples de tâches qui peuvent facilement être automatisées par le biais de programmes.

### III.3. Système de gestion de Workflow (SGWF)

Un système de gestion de workflow est un système qui définit, crée et gère l'exécution de workflows grâce à l'utilisation d'un logiciel capable d'interpréter les définitions de processus, d'interagir avec les participants et, lorsque cela est requis, d'invoquer les outils et les applications. (Fdhila, 2011a)

D'après (Gaaloul., 2006) "*Les systèmes de gestion de workflows sont des collecticiels<sup>2</sup> qui offrent des mécanismes de modélisation et d'automatisation de processus métiers. Ils sont parmi les systèmes les plus élaborés pour définir et exécuter des processus. Ils permettent, en particulier, de décrire explicitement les méthodes de travail réalisant un processus, de les expérimenter et de mesurer leurs qualités.*"

#### III.3.1. Modèle de référence de Workflow

Pour réaliser l'interopérabilité entre les différents produits de Workflow, il faut définir des standards d'interfaces d'échanges de données. WfMC a créé le *Modèle de référence du Workflow*. C'est un modèle qui décrit le système de gestion de workflow, porté sur la normalisation des interfaces entre ses modules, et de leurs clients. (Marek, 2006)

La Figure 7.1 illustre brièvement les composants de base du modèle de référence d'un SGWf ainsi que les interfaces entre ces composants comme définis par WfMC.

---

<sup>2</sup> Collecticiels correspond au terme "groupware" qui représente un type de logiciel qui permet à un groupe de personnes de partager des documents à distance pour favoriser le travail collaboratif.

# Partie 01: Etat de l'art et Problématique

## Chapitre 01: Workflow

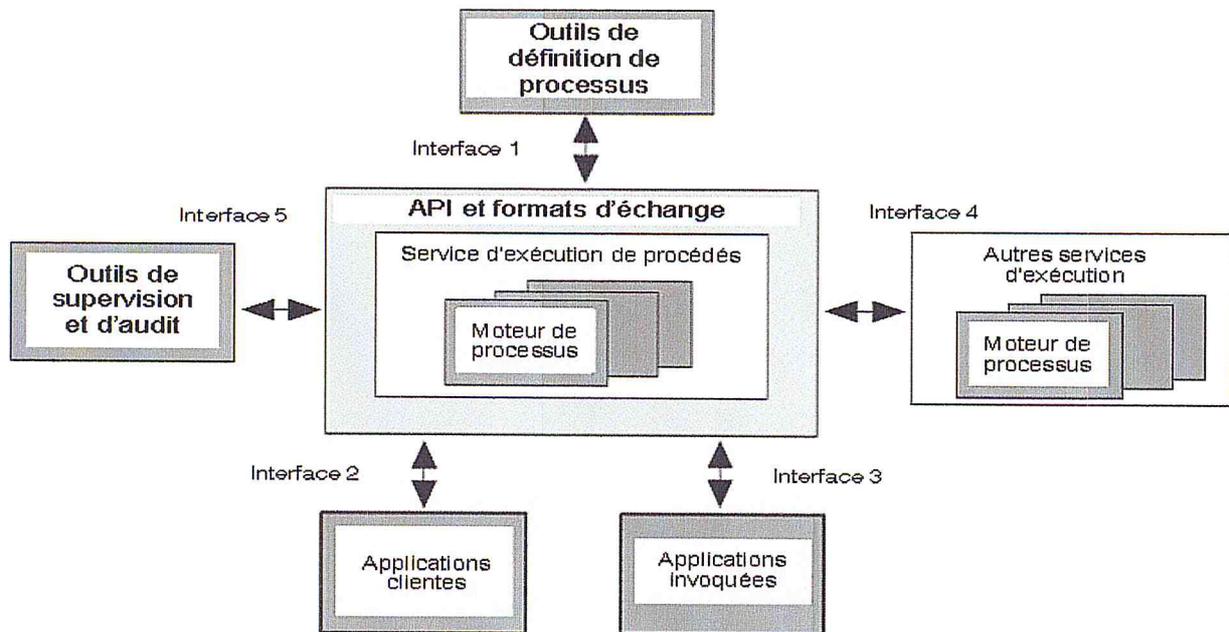


Figure 8.1: Le modèle de référence du Workflow (Benfifi, 2012a)

- **Le service d'exécution des processus:** est un service logiciel composé d'un ou plusieurs moteurs Workflow de même type qui servent à définir, gérer et exécuter des Workflow.
- **Le moteur de processus:** est un service logiciel qui fournit tout ou partie de l'environnement d'exécution d'un workflow. (Benfifi, 2012b)
- **Applications clientes:** fournit une interface aux utilisateurs qui participent à l'exécution du Workflow, elles comprennent la liste des tâches, dont la tâche principale est de fournir, aux participants au workflow des objets qui peuvent être exécutés dans le processus. (Jonuzi, 2014c)
- **Applications invoquées:** l'exécution d'un processus pourrait nécessiter l'invocation d'une autre application externe au processus Workflow. L'application appelée est un participant au système. les applications invoquées incluent les applications interactives ainsi les applications automatisées. (Jonuzi, 2014d)
- **Autres services d'exécution:** les systèmes de gestion de flux de travail peuvent se composer de plusieurs moteurs de Workflow. Dans ce cas, un objectif clé de la WfMC est de définir des normes qui permettront aux systèmes de Workflow édités par différents fournisseurs, pour se passer des éléments de travail de manière transparente entre eux, par le biais de l'interface 4. (Jonuzi, 2014e)
- **Outils de supervision et d'audit** Ces outils permettent généralement aux administrateurs, de gérer le système de gestion de Workflow. Ils permettent l'ajout ou la suppression des ressources, ainsi que la présentation de données statistiques sur les processus. (Jonuzi, 2014f)

#### Interfaces du SGWf

- **L'Interface 1** fournit le lien entre les outils de création et de modification de workflows et le moteur de workflow. **(Benfifi, 2012b)**
- **L'Interface 2** permet la communication entre les applications clientes du workflow et le moteur de workflow. **(Benfifi, 2012c)**
- **L'Interface 3** permet d'invoquer des applications bien déterminées d'une activité donnée afin d'exécuter des tâches spécifiques. **(Benfifi, 2012d)**
- **L'Interface 4** permet l'interopérabilité et l'échange de travail entre plusieurs systèmes de gestion de workflows autonomes. **(Benfifi, 2012e)**
- **L'Interface 5** permet l'interconnexion entre les outils d'administration et de surveillance et le moteur de workflow. Elle est divisée en deux parties : les fonctions de système de gestion de workflow et les fonctions de cheminement de workflow. Les outils d'administration et de surveillance peuvent auditer par exemple les temps d'attente, de terminaison et d'exécution ainsi que le routage. **(Benfifi, 2012f)**

#### **III.4. Workflows inter- organisationnels**

Les Workflows que nous appelons processus métiers, ont pris une place prépondérante dans les entreprises durant ces dernières années grâce à leurs bénéfiques remarquables. Face à la concurrence et dans le but d'améliorer leur productivité, les entreprises expriment un grand besoin d'ouverture et de coopération à l'échelle mondiale. Elles ont besoin de s'allier à d'autres entreprises, de compétences complémentaires, afin de coopérer avec elles, et de réaliser des projets qui ne sont pas à la portée d'une seule entreprise, par exemple, externalisations intensives de processus et de services métiers. **(Chebbi, 2007)**

Dans ce sens on parle de Workflow Inter-Organisationnels (WFIO):

Le WFIO est considéré comme une extension du workflow traditionnel puisqu'il envisage la coopération de plusieurs processus issus de plusieurs organisations réparties, autonomes et hétérogènes. 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.

Il se distingue du Workflow traditionnel par trois aspects essentiels :

- La répartition des processus d'organisations.
- L'autonomie des organisations : chaque organisation, prise individuellement, décide par elle-même les conditions de coopération, c'est à dire quand, comment et avec qui elle doit coopérer.
- L'hétérogénéité des organisations à faire coopérer : cela concerne les différences en termes de modèles et de systèmes. **(Benfifi, 2012g)**

#### ❖ **Problèmes liés au WFIO**

## Partie 01: Etat de l'art et Problématique

### Chapitre 01: Workflow

Lors d'une collaboration entre les entreprises en question, dans le cadre par exemple d'un scénario B2B « Business-to-Business »<sup>3</sup> on met en correspondance les activités correspondantes ce qui nécessite de rechercher des interfaces compatibles; et c'est là où réside la difficulté, parce qu'on doit réaliser une vérification non seulement de la compatibilité syntaxique (type et format des données) lors de l'échange des informations entre les organisations collaboratrices mais aussi une vérification de la compatibilité sémantique des concepts utilisées entre les différents partenaires.

#### III.5. Langages de modélisation de Workflow

La modélisation des processus métier est au cœur même de la démarche d'analyse dynamique d'une organisation. Que ce soit dans le cadre d'une démarche d'amélioration ciblée ou d'une réorganisation plus globale, la modélisation des processus permet de formaliser le fonctionnement précis d'une organisation en utilisant un langage standard et aisément compréhensible. Dans ce sens, plusieurs langages ont été proposés dans la littérature.

##### III.5.1. EPC (Event-driven Process Chain)

Le Event-driven Process Chain (EPC) est un langage de modélisation pour la représentation graphique des processus d'affaires des organisations. Dans le cadre de l'architecture des systèmes d'information intégrés, cette méthode a été développée comme une approche holistique pour la modélisation de vue orientée de processus d'affaires.

L'EPC fournit des processus d'affaires dans un langage de modélisation semi-formel avec des règles de syntaxe d'un point de vue centré sur les événements. En principe, il est possible d'exprimer le contrôle et la circulation de l'information d'un processus d'affaires.

**(Devillers, 2011)**

La Figure ci-dessous représente un exemple d'un Workflow en EPC.

---

<sup>3</sup> B2B: désigne l'ensemble des activités commerciales nouées entre deux entreprises, et plus largement les moyens techniques utilisés pour mettre en relation ces entreprises et faciliter leurs échanges de produits, de services ou d'informations.

## Partie 01: Etat de l'art et Problématique

### Chapitre 01: Workflow

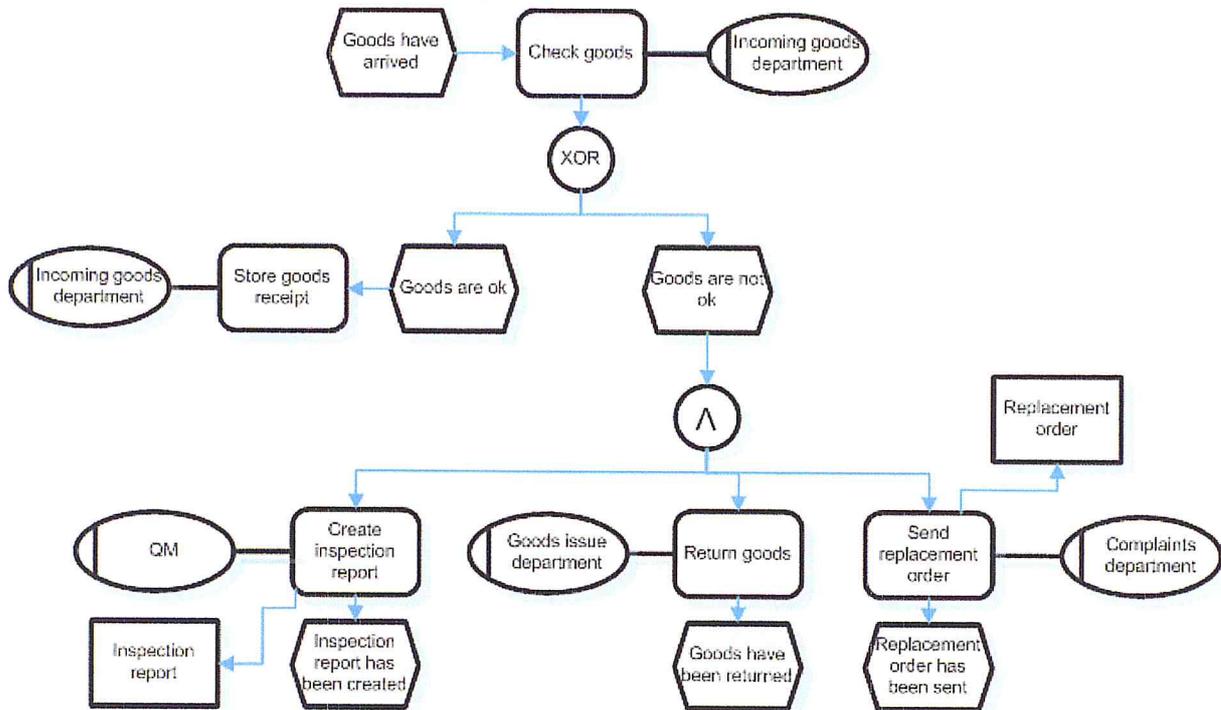


Figure 1.2: Exemple d'un Workflow en EPC. (bpm-book)

### III.5.2.UML (Unified Modeling Language)

UML (*Unified Modeling Language*) est un langage de modélisation orientée objet développé en réponse à l'appel à propositions lancé par l'OMG (*Object Management Group*) dans le but de définir la notation standard pour la modélisation des applications construites à l'aide d'objets. (Dardek & Zouag, 2011)

Il définit des diagrammes structurels et comportementaux, pour représenter respectivement des vues statiques et dynamiques d'un système :

- diagrammes de classe,
- diagramme de cas d'utilisation,
- diagramme de séquence ..etc.

Mais le diagramme le plus important dans UML pour modéliser des processus métier est le diagramme d'activités, car il a un large éventail d'utilisations, en ce qu'il peut montrer (activités séquentielles et en parallèle), les objets consommés, utilisés ou produits par une activité, les acteurs responsables d'une activité, et les relations et dépendances entre activités. Tout cela est essentiel dans la modélisation d'entreprise.

La Figure suivante représente un exemple d'un diagramme d'activité en UML qui modélise le fonctionnement d'une borne bancaire :

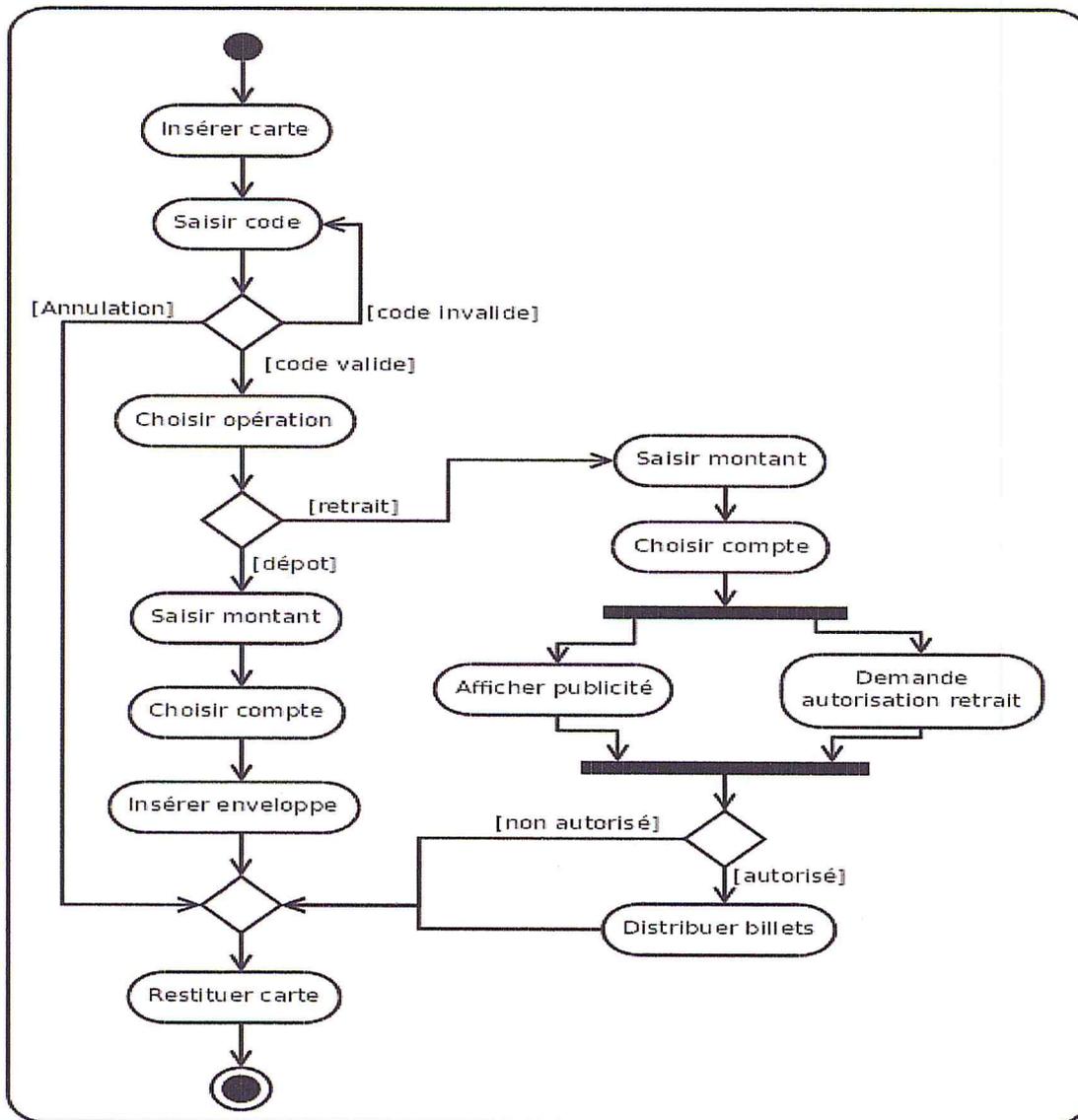


Figure 1.3: Diagramme d'activité en UML. (developpez.com)

### III.5.3.BPMN

La Business Process Management Initiative (BPMI), organisme qui anime un courant de standardisation dans le domaine du management par les processus métiers, a mis au point une norme de notation de modélisation des processus métier; la "Business Process Model and Notation (BPMN)". (Touzi, 2007)

BPMN est maintenant maintenu par l'OMG «Object Management Group», qui l'a adopté comme norme, qui est devenue aussi intéressante pour entreprises.

La spécification BPMN 1.2 a été publiée en janvier 2009 et la nouvelle version BPMN 2.0 a été publiée en 2011.

L'objectif principal de l'effort BPMN était de fournir une notation qui soit facilement compréhensible par tous les utilisateurs métiers, depuis les analystes qui créent les premières ébauches des processus, jusqu'aux développeurs techniques responsables de l'implémentation de la technologie qui va s'acquitter de ces processus, et, enfin, les gens du métier qui vont gérer et contrôler ces processus (ElMansouri, 2011)

## Partie 01: Etat de l'art et Problématique

### Chapitre 01: Workflow

La notation BPMN, décrivant statiquement les processus, a permis d'améliorer les possibilités des notations traditionnelles en gérant les concepts de procédures B2B, comme les processus publics et privés, et les chorégraphies (à partir de sa version 2.0), ainsi que des concepts de modélisation avancée, comme la gestion des exceptions et la compensation des transactions. **(Baouab, 2013b)**

On a bien détaillé ce langage dans l'annexe B de ce mémoire.

#### III.6. Langages d'exécution de Workflow

Beaucoup de langages de spécification de processus métiers sont apparus ces dernières années (XPDL, WSFL, XLANG, etc.). Chacun adopte sa terminologie particulière et définit ses concepts spécifiques en fonction des objectifs visés par le métamodèle de Workflow, du domaine d'intérêt ou du choix de représentation. Ces spécifications sont techniques et sont souvent basées sur le langage XML.

##### III.6.1. BPEL (Business Process Execution Language)

Initialement connu sous le nom BPEL4WS (Business Process Execution Language For Web Service), renommé par la suite WS-BPEL (Web Services Business Process Execution Language), cette spécification est plus connue sous le nom BPEL.

Le standard BPEL est une spécification proposée conjointement par IBM, Microsoft, et BEA, et représente une convergence des idées initialement proposées par les langages XLANG et WSFL. **(Fdhila, 2011b)**

C'est aussi un langage pour définir et gérer les activités d'un processus métier. Ce langage permet de décrire des protocoles d'interactions et de collaborations entre les services Web sur lesquels s'appuie le processus. BPEL utilise le modèle de contrôle classique des flots de tâches (Workflows) pour décrire des processus métiers invoquant des services Web. **(Helga, 2007)**

BPEL supporte les processus exécutables et abstraits, et ce, grâce à son mécanisme générique de spécification de concepts.

##### III.6.2. XPDL

XPDL (XML Process Definition Language) est un langage proposé par la WfMC (Workflow Management Coalition) qui permet de définir des processus et utilise le langage XML comme support. XPDL est un méta modèle de définition de processus. Ce dernier se positionne comme un format d'échange entre des outils de modélisation et des moteurs d'exécution de Workflows. **(Zefouni, 2012)**

XPDL est l'implémentation de Interface 1 du modèle de référence de Workflow. C'est aussi un langage suffisamment large pour qu'il puisse être adopté par différents vendeurs de systèmes de gestion de Workflow. **(Franck, s.d.)**

Il représente un standard supportant tous les éléments pour la définition de Workflows tels que les flots de contrôle, les flots de données, la gestion des ressources et des applications. **(Nomane, 2008)**

Pour mieux comprendre son utilité ainsi sa syntaxe vous pouvez consulter l'annexe C de ce mémoire.

#### III.7. Editeurs Workflows

Ils permettent de modéliser et d'automatiser les processus métier de l'entreprise. Dans ce qui suit, une liste des éditeurs de Workflow open-source les plus connus :

##### III.7.1. ORYX Editor

Oryx est le résultat du projet académique « Browser-based Business Process Editor » initié par le département « Business Process Technology » de l'institut Hasso-Plattner (aujourd'hui il est président du conseil de supervision et n'a donc plus un rôle opérationnel dans les activités de l'entreprise) à l'université de Postdam en Allemagne, il a été supervisé par le professeur Mathias Weske, responsable du groupe de recherche des processus d'affaires.

L'objectif du projet était de développer une application Web pour la modélisation des processus d'affaires qui permettrait de supporter plusieurs notations graphiques standards : BPMN, EPC, Réseaux Pétri, etc. C'est en 2007 qu'Oryx a été déployé et mis à la disposition des utilisateurs et des développeurs des logiciels libres. (Belghait, 2011)

##### III.7.2. Activiti

Activiti est développé, sous l'égide d'Alfresco (est une plateforme ouverte permettant la gestion de documents critiques des entreprises). Distribué sous licence Apache, c'est un produit sérieux et performant, en plein essor sur le marché des BPMS Open Source.

Activiti fournit deux composants différents pour la modélisation des processus d'affaires. Activiti Modeler est une application Web, tandis que Activiti designer est un plugin dédié pour l'IDE Eclipse (Platform est principalement écrit en Java).

Activiti Modeler est une version personnalisée de l'éditeur de processus Signavio (on a vu entamé dans la section suivante), il est utilisé pour la création des diagramme BPMN 2.0 à l'aide d'un navigateur. Les fichiers de processus sont stockés par le serveur dans un référentiel de modèle de base de données. Depuis Activiti 5.7, il a été déplacé vers un projet distinct, qui est appelé Signavio Core Components.

##### III.7.3. Signavio Core Components

Signavio Core Components permet la création de base et l'échange de diagrammes BPMN 2.0. Il a un backend basé sur des fichiers inclus dans le référentiel.

Le projet Signavio Core Components contient les bases techniques pour éditer des diagrammes de processus.

Le développement de Signavio Core Components a été arrêté en 2011, et son code n'est plus maintenu. Aujourd'hui Signavio a aussi proposé une version extensible. Ce logiciel est beaucoup plus facile à utiliser et à configurer.

5% de Signavio est basé sur Oryx alors que 95% est un code source fermé. Le logiciel est disponible comme une plateforme Cloud de type Software-as-a-Service (SaaS) ou en achat comme licence "On-premise" (installée chez l'utilisateur).

## Partie 01: Etat de l'art et Problématique

---

### Chapitre 01: Workflow

Il offre une version QuickModel pour les débutants basée sur les feuilles de calcul avec une génération automatique de BPMN 2.0. (**signavio-core-components**)

#### IV. Conclusion

Ce chapitre nous a permis de faire une analyse globale et une étude précise sur les meilleurs mécanismes utiles au bon fonctionnement d'une entreprise. Autrement dit comment bien automatiser les processus métier c'est-à-dire bien décrire les workflows syntaxiquement, y compris leurs composants, modélisation et exécution.

Dans le chapitre suivant, nous allons montrer l'importance d'un autre type de description pour les workflows, notamment les workflows inter-organisationnels, qui est la description sémantique qui permet de mettre en adéquation les entreprises collaborant.



***Chapitre 02***  
***Description sémantique des***  
***Workflows***

## I. Introduction

Un workflow correspond à une composition de programmes informatiques et particulièrement à une composition de services Web. Aussi les workflows sont particulièrement utilisés pour décrire comment des services ou activités "informatiques" sont connectés pour construire des procédés métiers complexes.

L'objectif de ce chapitre est de définir précisément ces notions ainsi que de couvrir tout ce qui en dépend, tel que les langages utilisés, ainsi que l'intérêt important de la description sémantique de ces workflows.

## II. Les services Web

Les services Web sont des programmes informatiques permettant la communication entre applications distantes à travers le réseau Internet indépendamment de tout langage de programmation et de toute plate-forme d'exécution, sans intervention humaine, et en temps réel. Les acteurs participant à une communication de services Web sont (voir figure ci-après):

- Le fournisseur du service qui se charge de la description des messages manipulés et des profils des fonctionnalités offertes par le service. Il doit publier le service Web créé dans un annuaire afin qu'il puisse être trouvé par des clients ;
- L'annuaire de services fournit des informations sur la description et la localisation des services Web ;
- Le client est une application qui cherche, localise et invoque le service.

Les services Web s'appuient sur un ensemble de protocoles standardisant les modes d'invocation entre les acteurs participant à la communication. Il s'agit de SOAP jouant le rôle de protocole de communication, de WSDL permettant la description de services et d'UDDI jouant le rôle d'annuaire de services. (Belaid, 2011a)

Figure 9.1 représente le mécanisme de fonctionnement de service web.

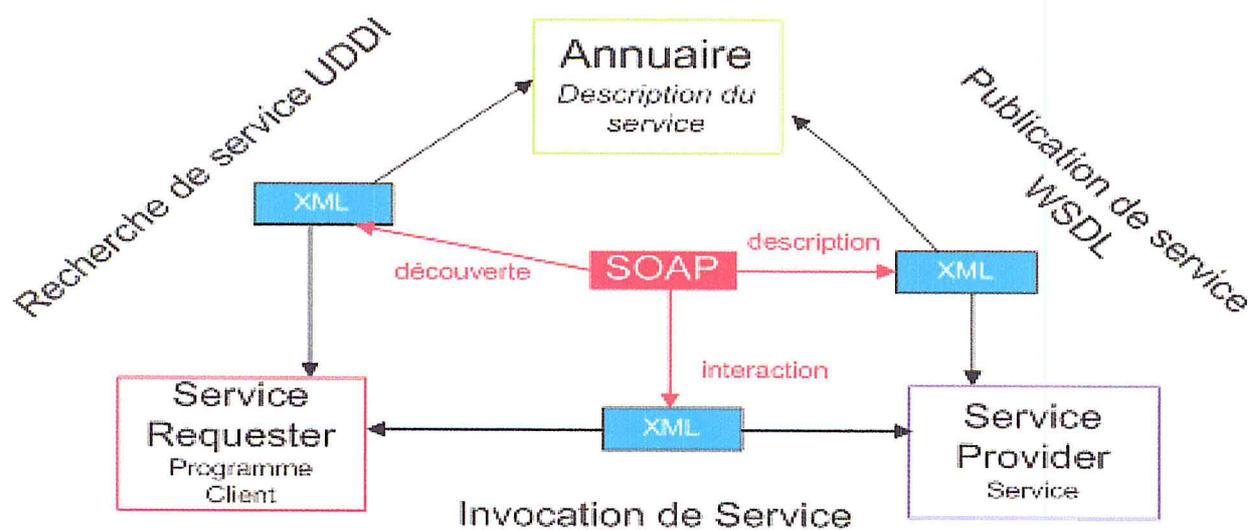


Figure 9.1: Fonctionnement des services Web (Death, 2011a)

### II.1. Web Service Description Language (WSDL)

WSDL (*Web Services Description Language*) est un langage de description standard, Il a été initialement développé par Microsoft et IBM et soumis au W3C devant une commission de 25 compagnies. C'est l'interface présentée aux utilisateurs. Il indique comment utiliser le service Web et comment interagir avec lui. WSDL est basé sur XML et permet de décrire de façon précise les détails concernant le service Web tels que les protocoles, les ports utilisés, les opérations pouvant être effectuées, les formats des messages d'entrée et de sortie et les exceptions pouvant être envoyées. (Death, 2011b)

### II.2. Simple Object Access Protocol (SOAP)

C'est un protocole de dialogue par appels de procédures à distance entre objets logiciels. Sa syntaxe d'utilisation est fondée sur XML et ses commandes sont envoyées sur Internet par l'intermédiaire du protocole HTTP mais aussi SMTP et POP sous forme de texte structuré. Il permet aux systèmes à objets distribués de solliciter et d'obtenir des services rendus par d'autres objets, il est moins lourd à mettre en œuvre que d'autres protocoles et c'est pour cela qu'il est de plus en plus adopté. (BOUHAJJA, 2012)

C'est l'épine dorsale du système d'interopérabilité. SOAP est un protocole décrit en XML et standardisé par le W3C. Il se présente comme une enveloppe pouvant être signée et pouvant contenir des données ou des pièces jointes.

Par rapport à tous les autres protocoles, SOAP est inter opérable, ainsi il est indépendant des plates-formes et langages de programmation. L'autre avantage réside dans le déploiement des applications.

### II.3. Universal Description, Discovery and Integration (UDDI)

UDDI (*Universal Description, Discovery and Integration*) est un annuaire de services. Il fournit l'infrastructure de base pour la publication et la découverte des services Web. UDDI permet aux fournisseurs de présenter leurs services Web aux clients.

UDDI se comporte lui-même comme un SW dont les méthodes sont appelées via le protocole SOAP.

Les informations qu'il contient peuvent être séparées en trois types :

- les pages blanches qui incluent l'adresse, le contact et les identifiants relatifs au service Web ;
- les pages jaunes qui identifient les secteurs d'affaires relatifs au service Web ;
- les pages vertes qui donnent les informations techniques. (Death, 2011b)

### III. La sémantique dans les services Web et les Workflows

La composition des services Web ainsi que la construction de workflows à partir de services permettent la réalisation des tâches complexes. Toute fois, seuls les informations utiles à leur gestion informatique sont spécifiées dans des langages de description de service ou de workflow tel que XPD, WSDL et BPEL. Ces descriptions, essentiellement syntaxiques, ne sont pas suffisamment riches pour permettre une gestion complexe supportant la découverte, l'interopérabilité ou l'adaptation au niveau sémantique. Et donc l'absence de la sémantique explicite rend les services et workflows informatiques difficilement compréhensibles, réutilisables et composables pour les utilisateurs ayant des connaissances techniques et informatiques limitées. (Belaid, 2011b)

Dans ce contexte, on va souligner dans les sections suivantes l'importance de l'aspect sémantique dans les workflows ainsi que les services Web.

### IV. L'annotation sémantique

#### IV.1. Qu'est-ce qu'une annotation

Les annotations sont considérées comme des métadonnées particulières en ce sens qu'elles représentent de nouvelles données attachées à des ressources documentaires. (Hernandez, 2005) L'annotation n'est pas une fin en soi, mais elle a pour but d'être réutilisée ultérieurement. Plusieurs auteurs ont établi les objectifs précis liés aux annotations (Mille, 2005) tels que classifier, hiérarchiser, reformuler, etc.

Dans la Figure 2.2: Annotation linguistique d'un document.2, nous présentons un exemple d'annotation de documents réalisée dans le cadre du projet WokHub (Consortium, 2008) qui a pour objectif l'annotation de documents issus de la géologie pétrolière. Dans cet exemple, ils ont montré l'une des première étapes d'annotation qui a pour but la reconnaissance de la langue d'un document.

Table 1. Work-packages of the PICOREF project Storage capacity assessment Selection of favourable areas for CO2 injectivity Source to sink matching Guaranty of reservoir integrity and its cap rock Development of methods and Improvement tools for modelling the short- and long- term behaviour of the system. Development of risk assessment following methodology is applied (table 2): Table 2. Methodology (PICOREF project) head2right Description of of phenomena involved head2right Risk assessment head2right Tools and methodology for control head2right Res checkblid Existing regulations checkblid Lack of regulations checkblid Documentation required by French adminis PRELIMINARY RESULTS The characterisation of the reservoir and cap-rock geometry was based on the reproac acquired during the last three decades. Specific processing was carried out so as to increase the quality of re corrections were applied to correct the effect of the superficial Tertiary formations and the lateral Chalk (13-layer model). Six regional transects were laid out and have allowed the construction of a geometric n important that internal reservoir structures were identified (prograding wedges). Two targets are identif trap associated with one of the major faults in the Paris basin (the Saint Martin de Bossenay fault) and



Annotation de la langue du document et génération d'un fichier XML contenant le document Initial

```

</annotation>
<mediaUnit xmlns:ns4="http://weblab-project.org/core/model/text" xsi:type="ns4:text" uri="weblab://myrepository/d
<annotation uri="weblab://myrepository/doc_090320_163646#0-a0">
  <data xmlns:ns5="http://weblab-project.org/core/model/" xmlns:ns6="http://weblab-project.org/core/model/t
  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:dc="http://purl.org/dc/element
    <rdf:Description rdf:about="weblab://myrepository/doc_090320_163646#0">
      <dc:language>en</dc:language>
    </rdf:Description>
  </rdf:RDF>
</data>
</annotation>
<segment xsi:type="ns4:linearSegment" uri="weblab://myrepository/doc_090320_163646#0_inriaPorP05_1">
  <start>1002</start>
  <end>1095</end>
</segment>
<content>
Table 1. Work-packages of the PICOREF project Storage capacity assessment Selection of favourable areas for C
injectivity Source to sink matching Guaranty of reservoir integrity and its cap rock Development of methods a
Injectivity Improvement tools for modelling the short- and long- term behaviour of the system. Development of
above, the following methodology is applied (table 2): Table 2. Methodology (PICOREF project) head2right
Characterisation of phenomena involved head2right Risk assessment head2right Tools and methodology for cont
Responsibilities of each actor PRELIMINARY RESULTS The characterisation of the reservoir and cap-rock g
onshore seismic profiles (800 km) acquired during the last three decades. Specific processing was carrie
particular, the necessary static corrections were applied to correct the effect of the superficial T
of sonic speed in the Mesozoic Chalk (13-layer model). Six regional transects were laid out and have all
increase in quality was so important that internal reservoir structures were identified (prograding wedg
  
```

Figure 2.2: Annotation linguistique d'un document.(Belaid,2011c)

## IV.2. Annotation sémantique

Les annotations sémantiques ont pour objectif d'exprimer la "sémantique" à l'intérieur d'une ressource afin d'en améliorer la compréhension, la recherche et donc la réutilisation par d'autres utilisateurs. (Corcho, 2006)

Le mécanisme d'annotation sémantique permet ainsi de lier une donnée à sa description sémantique représentée par un concept issu d'une ontologie. ce mécanisme est utilisé pour l'enrichissement sémantique des programmes et processus informatiques.

## V. Annotation sémantique des langages de workflows

### V.1. Semantic Annotation for WSDL (SAWSDL)

Est un langage sémantique de description de service Web. Il est évolutif et compatible avec les standards des services Web existants, et plus spécifiquement avec WSDL. D'une part, SAWSDL, fournit un mécanisme permettant d'annoter sémantiquement les types de données, les opérations, les entrées et les sorties de WSDL et d'autre part, il ajoute des éléments pour spécifier les pré-conditions, les effets et les catégories des services Web. (Chabeb, 2011)

Dans la Figure 2.3 nous présentons un exemple d'annotation du message produit en sortie par l'opération *devis* en l'associant au concept *DevisDeclaration* dans l'ontologie *AssuranceOnto*.

```

1 <wsdl:operation name="devis"
2   psawSDL: preference="http://w3/ontology/AssuranceOnto/Operation#Devis"/>
3   <wsdl:output element="devisDeclaration"
4   psawSDL: preference="http://w3/ontology/AssuranceOnto#DevisDeclaration"/>
5 </wsdl:operation>

```

Figure 2.3: Description de l'opération devis de l'assuré en SAWSDL.(Belaid,2011d)

Aujourd'hui SAWSDL est utilisé aussi pour annoter n'importe quel schéma XML et pas seulement WSDL, mais son inconvénient est la nécessité de modifier les outils existants autour de WSDL pour les adapter à SAWSDL même s'il a été conçu dans la perspective d'être un langage incrémental . Une autre limitation est qu'il n'aborde pas la composition de services.

## V.2. Semantic Service Markup ( SESMA )

SESMA est un langage de description de services Web sémantiques, qui a été conçu pour fournir un langage simple à utiliser, avec une syntaxe compacte et une intégration forte avec les langages existants WSDL, SOAP et WS-BPEL. SESMA a été construit avec les objectifs suivants :

- une syntaxe reposant sur le langage XML, pour une distribution du langage à large échelle ;
- une sémantique précise qui comporte la signification des descriptions ;
- une forte complémentarité avec les langages existants ;
- des possibilités d'extension permettant une évolution vers d'autres langages éventuels.

L'exemple représenté dans la Figure 2.4: Description SESMA pour l'annotation sémantique d'une opération WSDL., illustre un fragment de description SESMA pour l'annotation sémantique de l'opération WSDL *buyItem*.

Dctypes	http://purl.org/dc/dcmitype/	Dublin Core Type Vocabulary
Foaf	http://xmlns.com/foaf/0.1/	Vocabulaire friend-of-a-friend (FOAF)
Prov	http://www.w3.org/ns/prov#	Ontologie de provenance
Rdf	http://www.w3.org/1999/02/22-rdf-syntax-ns#	RDF
Rdfs	http://www.w3.org/2000/01/rdf-schema#	RDFSchema
Skos	http://www.w3.org/2004/02/skos/core#	Simple Knowledge Organization System

Tab 2.1: Tableau représentant des namespaces définis par OA (openannotation)

### V.3.3. Les concepts de Body et Target

Typiquement une annotation a un seul Body, qui est soit un commentaire soit une autre ressource descriptive, et une seule Target ce que le Body est en quelque sorte "à propos". Le Body fournit l'information sur la Target de l'annotation. Le modèle de base est représenté par la Figure ci-dessous.

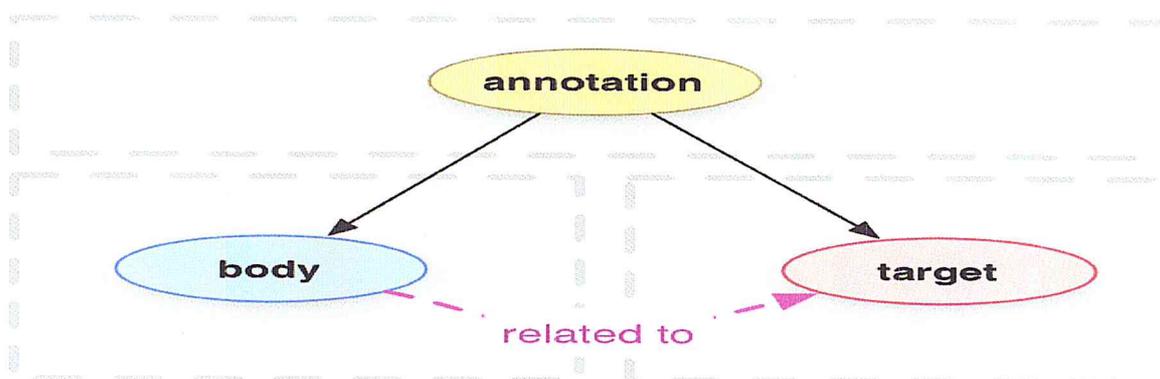


Figure 2.5: Les concepts « Annotation », « Body » et « Target » du modèle open annotation. (openannotation)

La nature exacte de cette relation change en fonction de l'intention de l'annotation, mais transmet le plus souvent ce que le Body est en quelque sorte "à propos de" la Target. D'autres relations possibles comprennent que le Body est un identifiant pour la Target, donnant une représentation de la Target, ou classification de la Target d'une certaine façon.

Le Body et la Target peuvent être de tout type de support, et contenir tout type de contenu. Ils doivent être identifiés par des URI, sauf s'ils sont intégrés au sein de l'annotation.

Toutes les annotations doivent être une instance de la classe: oa:Annotation. Le modèle définit deux relations, oa: hasBody et oa: hasTarget, pour associer les ressources du Body et de la Target, respectivement, avec l'annotation.

La Figure 2. suivante représente un modèle d'annotation de base.

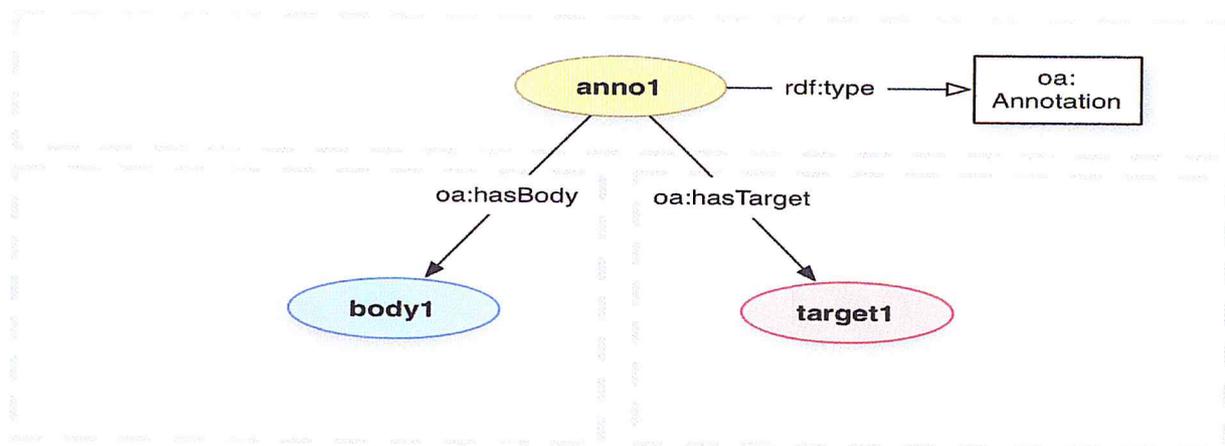


Figure 2.6: Modèle de base d'Open Annotation. (**openannotation**)

### V.3.4. Typage de Body et Target

Ce sont des informations concernant le type du contenu général (texte, image, audio, vidéo, etc.) des ressources et sont utiles pour les applications qui exploitent les annotations. Ceci est exprimé par le typage des Body et Target.

La Figure 2.7 représente un exemple de typage d'un Body et Target.

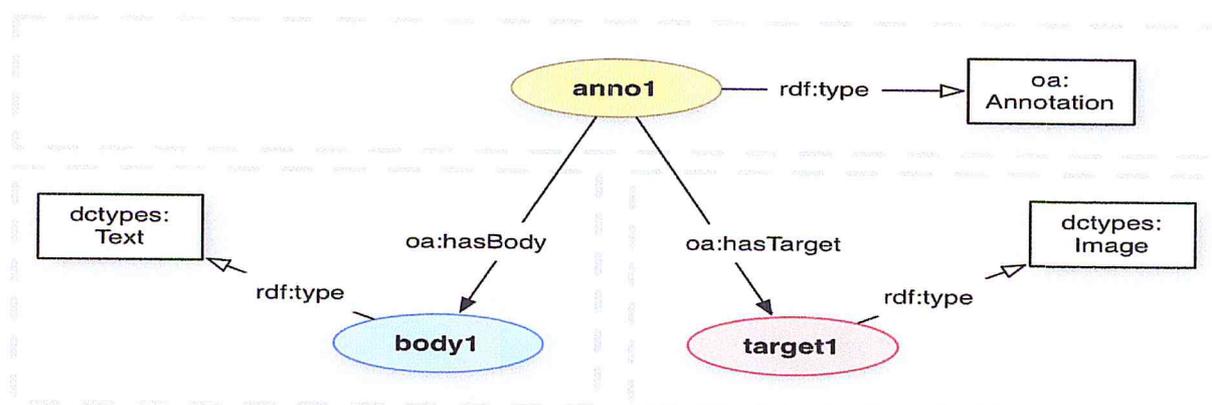


Figure 2.7: Typage d'un Body et Target. (**openannotation**)

### V.3.5. Motivation

Dans de nombreux cas, il est important de comprendre les raisons pour lesquelles l'annotation a été créée. Bien que les systèmes précédents ont sous classé la classe d'annotation pour transmettre ces motivations, il a été considéré qu'une description plus riche et mieux détaillée pourrait être obtenue en utilisant une hiérarchie SKOS.

SKOS est un modèle de données commun pour les systèmes d'organisation des connaissances tels que les thésaurus, systèmes de classification, systèmes de vedettes et des taxonomies. L'utilisation SKOS, peut être exprimée sous forme de données lisibles par la

machine. Elles peuvent alors être échangées entre les applications informatiques et publiées dans un format lisible par la machine dans le Web.

Chaque annotation devrait avoir au moins un oa:motivatedBy par rapport à une instance de la classe oa:Motivation, c'est une sous-classe (sub-class) de skos:Concept. Parmi les instances de la classe oa:Motivation qu'on peut utiliser lors de l'annotation: oa:classifying, oa:commenting, oa:describing, oa:tagging....etc. On a aussi la possibilité d'étendre cette classe pour définir d'autre type d'annotations comme des annotations sémantiques et contextuelles.

Le tableau présenté ci-dessous regroupe un ensemble des instances de la classe oa:Motivation.

Vocabulaire	Type	Description
oa: MotivatedBy	Relation	La relation entre une annotation et une motivation. Il devrait y avoir au moins une motivation pour chaque annotation, et peut être plus que 1.
oa: identifying	Instance	La motivation qui représente l'affectation d'une identité à la ressource (s) cible. Par exemple, annoter le nom d'une ville dans une chaîne de texte avec l'URL qui l'identifie.
oa: classifying	Instance	La motivation qui représente l'affectation d'un type de classification, généralement à partir d'un vocabulaire contrôlé, à la ressource (s) cible. Par exemple, pour classer une ressource image comme Portrait.
oa: commenting	Instance	La motivation qui représente un commentaire sur ou à l'examen de la ressource (s) cible. Par exemple, pour fournir un commentaire sur un fichier PDF particulier.
oa: describing	Instance	La motivation qui représente une description de la ressource (s) cible, par opposition à une observation à leur sujet. Par exemple décrivant les PDFs, plutôt que de commenter sur leur exactitude.
oa: editing	Instance	La motivation qui représente une demande de modification ou de modifier à la ressource cible. Par exemple, une annotation qui demande une faute de frappe d'être corrigée.
oa: tagging	Instance	La motivation qui représente l'ajout d'une marque sur la ressource (s) cible.

Tab 2.2: Tableau représente l'ensemble des instance de la classe oa. (**openannotation**)

La Figure 2.8: Exemple d'une motivation illustre un exemple d'une motivation.

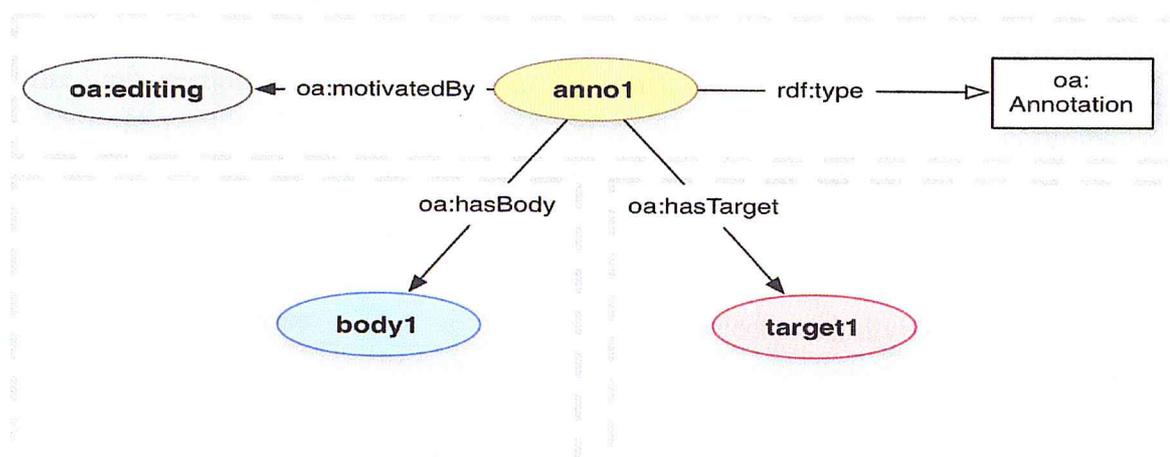


Figure 2.8: Exemple d'une motivation. (**openannotation**)

### V.3.6. Specifiers et Specific Resource

Les Specifiers sont associés directement avec chaque ressource spécifique. Dans ce contexte, la ressource complète est appelée ressource source, et donc Specifiers décrit comment déterminer les aspects de la Source qui constituent la ressource spécifique. Par exemple, une zone circulaire dans une image est identifiée par la ressource spécifique, décrite par un cercle, et la ressource de l'image complète est la Source. Les propriétés et les relations qui peuvent être exprimés sur le Body et la Target, telles que le type, le format et la provenance, doivent rester attachés à la ressource de Source. (**openannotation**)

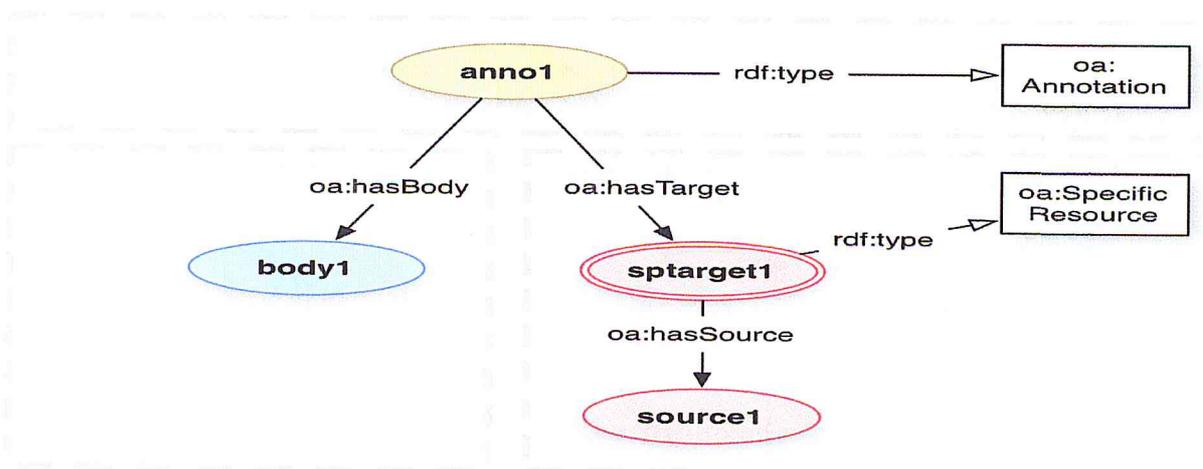


Figure 2.9: Specifiers et Specific Resource (**openannotation**)

### V.3.7. Selector

Beaucoup d'annotations ciblent une partie particulière de la ressource. Comme par exemple, des annotations sur une zone dans une image ou une vidéo, une plage de caractères dans le texte. De même, le segment peut être le corps de l'annotation, où le

identifiant de fragment de l'URI, et il est inclus dans le graphe d'annotation via la propriété (rdf:value). Ainsi, un fragment d'URI peut être reconstruit par la concaténation du (oa:hasSource ) URI de ressource, plus un «#», plus la valeur. Par exemple, si l'URI de la ressource était " http://www.example.com/image.jpg" et la propriété rdf:value était "xywh = 1,2,3,4", puis l'URI résultant serait "http: // www.example.com/image.jpg#xywh=1,2,3,4 ".

Les utilisateurs doivent traiter la valeur du sélecteur de fragments basé sur le standard à exprimer en utilisant dcterms:conformsTo. Si ce type n'est pas représenté, l'utilisateur doit utiliser le type de support de la ressource Source afin de déterminer le sens du fragment.

Le tableau suivant est model de fragment selector

Vocabulaire	Type	Description
oa:FragmentSelector	Classe	Une ressource qui décrit le segment d'intérêt dans une représentation, à l'aide de l'identificateur de composant de fragment d'un URI.
rdf:value	propriété	Le contenu de la composante d'identification de fragment d'un URI qui décrit le segment souhaité dans la ressource. Le oa: FragmentSelector DOIT avoir exactement 1 propriété rdf: value.
dcterms:conformsTo	Relation	Le sélecteur Fragment devrait avoir une relation dcterms:conformsTo avec l'objet étant la spécification qui définit la syntaxe du fragment.

Tab 2.4: Tableau représente un modèle de sélecteur de fragments.  
(openannotation)

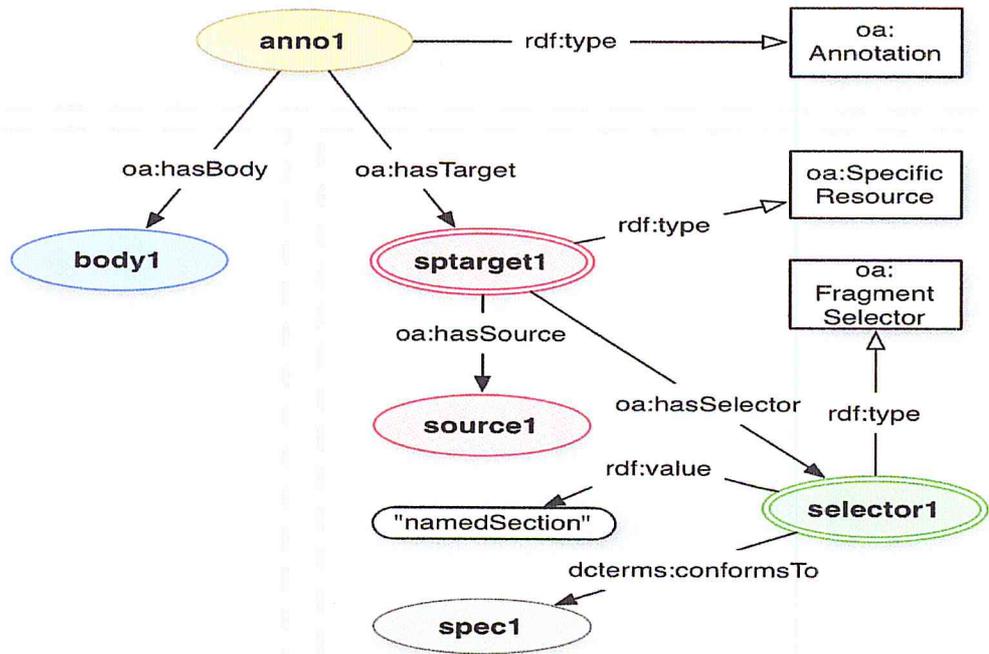


Figure 2.11: Fragment Selector. (openannotation)

### V.3.8. Sérialisation

Le format de sérialisation recommandée est JSON-LD. Cela doit permettre une implémentation simple au niveau des navigateurs Web. Pour consommer facilement les annotations en utilisant des outils et méthodes familières aux développeurs. Le contexte présenté ci-dessous est recommandé pour veiller à la cohérence entre les implémentations, et peut être référencé comme <http://www.w3.org/ns/oa-context-20130208.json>. Il est recommandé de supporter la négociation de contenu pour d'autres formats de sérialisation, y compris RDF/XML et Turtle. (openannotation)

Le résultat d'une sérialisation avec JSON-LD est comme ceci: (openannotation)

```
{
  "@context": "http://www.w3.org/ns/oa-context-20130208.json",
  "@id": "http://www.example.org/annotations/anno1",
  "@type": "oa:Annotation",

  "annotatedAt": "2012-11-10T09:08:07",
  "annotatedBy": {
    "@id": "http://www.example.org/people/person1",
    "@type": "foaf:Person",
    "mbox": {
      "@id": "mailto:person1@example.org"
    },
    "name": "Person One"
  },

  "hasBody": {
```

```

    "@id": "urn:uuid:1d823e02-60a1-47ae-ae7f-a02f2ac348f8",
    "@type": ["cnt:ContentAsText", "dctypes:Text"],
    "chars": "This is part of our logo"
  },
  "hasTarget": {
    "@id": "urn:uuid:cc2c8f08-3597-4d73-a529-1c5fed58268b",
    "@type": "oa:SpecificResource",
    "hasSelector": {
      "@id": "urn:uuid:7978fa7b-3e03-47e2-89d8-fa39d1280765",
      "@type": "oa:FragmentSelector",
      "conformsTo": "http://www.w3.org/TR/media-frags/",
      "value": "xywh=10,10,5,5"
    },
    "hasSource": {
      "@id": "http://www.example.org/images/logo.jpg",
      "@type": "dctypes:Image"
    }
  }
}

```

Le même exemple mais cette fois ci est représenté par RDF/XML

```

<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
  xmlns:oa="http://www.w3.org/ns/oa#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  <rdf:Description rdf:nodeID="Nfbb0cc8fea5b4182ba691d318d2da3e7">

  <rdf:type rdf:resource="http://www.exemple.org#Annotation"/>
  <oa:hasBody rdf:resource=" http://www.exemple.org/body1/>
  <oa:hasTraget rdf:resource=" http://www.exemple.org/target1/>

</rdf:Description>
</rdf:RDF>

```

un dernier exemple représenté avec RDF turtle : (openannotation)

```

<annol> a oa:Annotation ;
  oa:hasBody <body1> ;
  oa:hasTarget <sptarget1> ;
  oa:styledBy <style1> .

<style1> a oa:CssStyle, cnt:ContentAsText ;
  cnt:characterEncoding "utf-8" ;
  cnt:chars ".red { color : red }" .

<sptarget1> a oa:SpecificResource ;
  oa:hasSource <source1> ;
  oa:styleClass "red" .

```

### V.3.9. Objectifs du modèle

L'objectif principal du modèle de données Open Annotation est de fournir un mécanisme de description standard pour le partage entre les systèmes d'annotations. Cette interopérabilité peut être réalisée soit pour les partager avec d'autres, ou pour la migration des annotations privés entre les périphériques. Les annotations partagées doivent pouvoir être intégrées dans les collections existantes et réutilisées sans perte d'information significative. Le modèle devrait couvrir autant de cas d'utilisation d'annotation que possible, tout en conservant des annotations simples et faciles. (**openannotation**)

## VI. Conclusion

Dans ce chapitre, nous avons souligné les différents mécanismes et langages existants qui peuvent annoter les workflows et les services Web mais il faut bien dire que n'importe quelle annotation ne sera fonctionnelle et utile que si elle est associée avec des concepts d'une ontologie adéquatement choisie. Dans le chapitre suivant, on va réaliser une étude sur les ontologies.

*Chapitre 03*  
*Ontologie*

### I. 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, les WFIO 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, l'enseignement assisté par ordinateur, la découverte et la composition de services Web, etc.

Dans ce chapitre, nous allons introduire des généralités sur les ontologies, leurs composants, leurs langages spécifiques ainsi que les outils nécessaires et nous allons finir par la présentation d'un exemple d'ontologie.

### II. Les Ontologies

#### II.1. Définitions

**Ontologie** est un terme qui a tout d'abord été défini en Philosophie comme une branche de la métaphysique qui s'intéresse à l'existence, à l'être en tant qu'être et aux catégories fondamentales de l'existant. En effet, ce terme est construit à partir des racines grecques **ontos** qui veut dire ce qui existe, l'Être, l'existant, et **logos** qui veut dire l'étude, le discours, d'où sa traduction par *l'étude de l'Être* et par extension *de l'existence*. (Amardeilh, 2007)

Les ontologies sont tout simplement la description hiérarchique des concepts importants dans un domaine, couplé avec une description de chacun de ces concepts.

Selon (Gruber, 1993): « *Ontologie est la spécification explicite d'une conceptualisation. c'est la description des concepts et les relations qui peuvent être, existent pour un agent ou une communauté d'agent* ».

Selon (Farquhar, 1997) : « *Nous considérons ontologies toutes théories de domaine qui spécifient un vocabulaire spécifique de domaine d'entités, classes, propriétés, prédicats et fonctions, et d'être un ensemble de relations qui détiennent nécessairement parmi les termes de vocabulaire. Ontologies fournissent un vocabulaire pour représenter les connaissances sur un domaine et pour décrire des situations spécifiques dans un domaine* ».

En clair, une **ontologie** fournit les moyens d'exprimer les concepts d'un domaine en les organisant hiérarchiquement et en définissant leurs propriétés sémantiques dans un langage de représentation des connaissances formel favorisant le partage d'une vue consensuelle sur ce domaine entre les applications informatiques qui en font usage.

#### II.2. Composants d'une ontologie

Comme nous l'avons abordé, les ontologies fournissent un vocabulaire commun d'un domaine et définissent la signification des termes et des relations entre elles. La connaissance dans les ontologies est principalement formalisée en utilisant les cinq types de

## Partie 01: Etat de l'art et Problématique

### Chapitre 03: Ontologie

composants à savoir : **concepts** (ou classes), **relations** (ou propriétés), **fonctions**, **axiomes** (ou règles) et **instances** (ou individus). (LEFEBVRE, 2003)

- Les concepts, aussi appelés termes ou classe de l'ontologie, correspondent aux abstractions pertinentes d'un segment de la réalité (le domaine du problème) retenus en fonction des objectifs qu'on se donne et de l'application envisagée pour l'ontologie ;

Exemple:

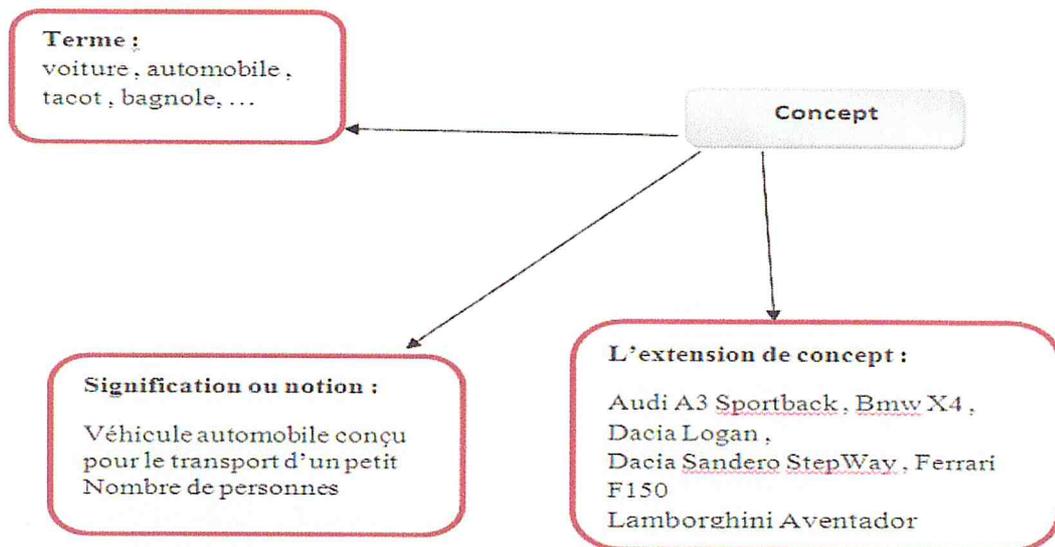


Figure 3.10: Exemple d'un concept (Benhbireche, 2014b)

- Les relations elles représentent des interactions entre concepts permettant de construire des représentations complexes de la connaissance du domaine. Elles établissent des liens sémantiques binaires, organisables hiérarchiquement. Ces relations incluent les associations suivantes :

- Sous classes de (généralisation-spécialisation) ;
- Partie de (agrégation ou composition) ;
- Associe à ;
- Instance de, etc.

Ces relations nous permettent d'apercevoir la structuration et l'interrelation des concepts, les uns par rapport aux autres.

- Les fonctions constituent des cas particuliers de relation, dans laquelle un élément de la relation, (le nième) est défini en fonction des N-1 éléments précédents ;
- Les axiomes les axiomes sont des expressions qui sont toujours vraies. Ils ont pour but de définir dans un langage logique la description des concepts et des relations permettant de représenter leur sémantique. Ils représentent les intentions des concepts et des relations du domaine et, de manière générale, les connaissances n'ayant pas un caractère strictement terminologique.

- Les instances constituant la définition extensionnelle de l'ontologie ; ces objets véhiculent les connaissances (statiques, factuelles) à propos du domaine du problème.

### II.3. Langages de représentation des ontologies

Plusieurs langages de spécification d'ontologies (ou langage d'ontologies) ont été développés pendant les dernières années. Certains d'entre eux sont basés sur la syntaxe de XML, tels que XOL (Ontology Exchange Language), SHOE (Simple HTML Ontology Extension qui a été précédemment basé sur le HTML), OML (Ontology Markup Language), RDF (Resource Description Framework), RDF Schéma et OWL (Web Ontology Language). (Zouhel, 2008a)

Figure 3.11: La pyramide des langages basés Web. représente les rapports principaux entre tous ces langages sous la forme d'une pyramide des langages du Web sémantique.

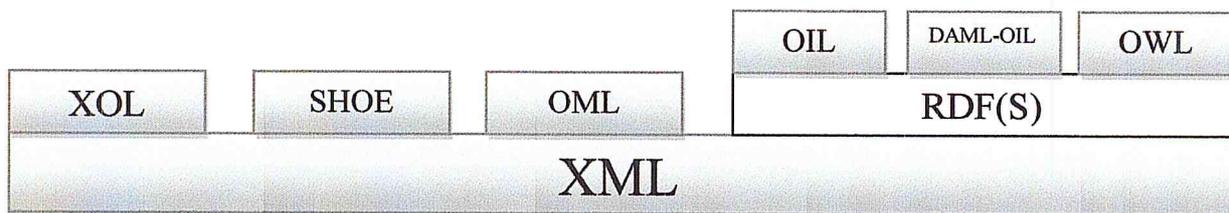


Figure 3.11: La pyramide des langages basés Web. (Zouhel, 2008b)

Dans cette section, nous présentons quelques langages de représentation des ontologies les plus connus et les plus utilisés.

#### II.3.1. RDF

Le langage Resource Description Framework (RDF) est le standard émergent proposé par le W3C pour la représentation et l'échange de métadonnées sur le Web, il est basé sur un modèle de triplets < sujet, prédicat, objet >, et possède une syntaxe XML.

- le sujet représente la ressource décrite, i.e. tout document accessible sur le Web comme les pages HTML, les documents textuels (PDF, Ms Word) ou multimédias (images, vidéo), etc., mais aussi tout objet, abstrait ou non, du monde réel. Les ressources sont nommées en utilisant une URI.
- le prédicat représente la propriété descriptive, i.e. une caractéristique spécifique, un attribut ou une relation, utilisée pour décrire une ressource.
- l'objet représente la valeur de cette propriété, soit une valeur littérale, comme un nombre entier ou une chaîne de caractères, soit une autre ressource accessible par son URI.

Voici un simple exemple d'une phrase composé des trois parties suivants:

- Sujet (Ressource) : index.html
- Prédicat (Propriété) : Titre

iii. Objet (valeur) : "ontologie"

#### II.3.2. RDFS

RDF Schéma (RDFS) est une recommandation du W3C en cours d'élaboration. Il s'agit du langage de spécification de schémas associé à RDF. Il est dédié à la représentation des connaissances ontologiques utilisées dans des déclarations RDF.

Un schéma RDF est un ensemble de déclarations de classes et de propriétés. Pour représenter les connaissances spécifiques à un domaine, les classes spécifiques au domaine sont déclarées comme des instances de la classe *Class* et les propriétés spécifiques au domaine comme des instances de la classe *Property*. Les propriétés *subClassOf* et *subPropertyOf* permettent de définir des hiérarchies de classes et des hiérarchies de propriétés.

#### II.3.3. OWL

OWL est un langage d'ontologie pour le web sémantique, développé par l'organisation W3C. Il facilite une grande interprétation du contenu du Web que celui fournit par XML, RDF ou RDF Schema (RDFS) en fournissant un vocabulaire riche avec une sémantique formelle. Il est basé sur la recherche effectuée dans le domaine de la logique de description, il repose sur la syntaxe des triplets RDF et réutilise certaines des constructions RDFS, il ajoute plus de vocabulaire pour décrire les propriétés et les classes. De plus, OWL tient compte de l'aspect diffus des sources de connaissances et permet à l'information d'être recueillie à partir de sources distribuées, notamment en permettant la mise en relation des ontologies et l'importation des informations provenant explicitement d'autres ontologies.

OWL a été fractionné en trois sous langages ou chacun est une extension par rapport à son prédécesseur :

- OWL Lite est destiné à être utilisé dans les situations où seulement une simple hiérarchie de classe et des contraintes simples sont nécessaires. Il a une complexité formelle inférieure à celle d'OWL DL. OWL Lite supporte seulement un sous-ensemble de constructions du langage OWL.
- OWL DL est beaucoup plus expressif que OWL Lite, il permet ainsi d'automatiser le calcul de la classification hiérarchique et la vérification des incohérences dans une ontologie OWL DL. Il inclut toutes les constructions du langage OWL, qui ne peuvent être utilisées que sous certaines restrictions.
- OWL Full est beaucoup plus expressif qu'OWL DL mais il ne permet pas d'automatiser le raisonnement au sein de l'ontologie. OWL Full permet à une ontologie d'augmenter la signification du vocabulaire prédéfini (RDF ou OWL). Il est peu probable que n'importe quel logiciel de raisonnement soit capable de supporter le raisonnement complet de chaque caractéristique d'OWL Full. Autrement dit, en utilisant OWL Full en comparaison avec OWL DL, le support

de raisonnement est moins prévisible puisque l'implémentation complète d'OWL Full n'existe pas actuellement.

#### II.4. Les éditeurs d'ontologie

Les éditeurs d'ontologie ont facilité la tâche du développeur de l'ontologie dans la construction des ontologies, en termes de définition des concepts du domaine, et de relations entre ces concepts dans la forme d'une hiérarchie de classes. Des exemples d'éditeurs d'ontologies sont : Protégé, SWOOP et Sémantic Works.

##### ➤ *Protégé*

Protégé est un logiciel gratuit (JAVA), plate-forme open source qui fournit une suite d'outils pour construire des bases de connaissance et des ontologies.

Protégé inclut de nombreux plugins pour la manipulation et la représentation d'ontologies dans différents formats. Il permet aussi de créer ou d'importer des ontologies écrites dans les différents langages d'ontologies tel que : RDF-Schéma, OWL, DAML, OIL, ...etc. Cela est rendu possible grâce à l'utilisation de plugins qui sont disponibles en téléchargement pour la plupart de ces langages. (Benhbireche, 2014)

L'éditeur Protégé-OWL permet aux utilisateurs de construire des ontologies pour le Web sémantique en OWL.

### III. SCOR (Supply Chain Operation Reference): ontologie de domaine pour les chaînes linguistiques

SCOR est un modèle de référence conçu pour une communication efficace entre les partenaires de la chaîne d'approvisionnement.

Le modèle SCOR a été conçu pour permettre aux entreprises de communiquer, comparer et apprendre des concurrents et des compagnies à l'intérieur et à l'extérieur de leur industrie. Il ne mesure pas seulement la performance de la chaîne d'approvisionnement, mais aussi l'efficacité de la chaîne d'approvisionnement. En outre, il a la capacité de tester et de planifier les futures améliorations de processus.

Le modèle SCOR fournit un cadre unique qui relie les processus d'affaires, les mesures, les meilleures pratiques de la technologie caractéristiques dans une structure unifiée pour soutenir la communication entre les partenaires de la chaîne d'approvisionnement et d'améliorer l'efficacité de la gestion de la chaîne d'approvisionnement et l'amélioration de la chaîne d'approvisionnement liées aux activités.

SCOR repose sur cinq processus de gestion distincts : la planification (plan), l'approvisionnement (source), la fabrication/transformation (make), la livraison (deliver), et le retour (return), comme le montre la Figure 3.3: .

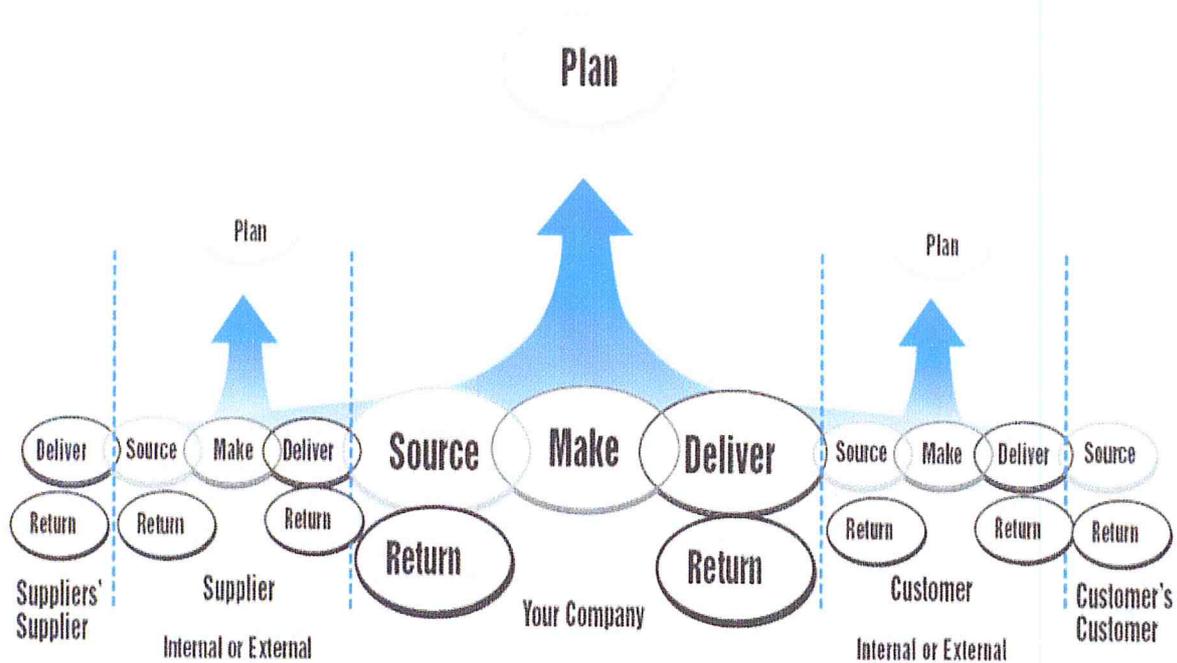


Figure 3.3: Processus de gestion de base du modèle de référence SCOR. (cat-logistique.com)

Donc, une ontologie, basée sur le fonctionnement SCOR modèle de référence et enrichie avec des significations sémantiques, va faire aligner la chaîne d'approvisionnement processus au sein des entreprises en réseau et faciliter l'interopérabilité.

#### IV. Conclusion

Avec ce chapitre nous sommes arrivées à réaliser une étude d'état de l'art sur les concepts et les notions nécessaires pour la réalisation de notre solution.

Dans le chapitre suivant, on va justifier et argumenter nos choix par rapport aux langages, techniques et mécanismes pour la réalisation de la solution proposée.



***Chapitre 04***  
***Choix techniques et***  
***méthodologiques***

### I. Introduction

Dans ce chapitre, nous allons exposer, à travers une étude comparative des différents outils et langages, nos analyses de choix techniques et méthodologiques pour la réalisation de notre objectif d'annotation sémantique des workflows inter-organisationnels, basée sur open annotation,

### II. Choix de l'outil approprié

Bien sûr, il serait difficile de chercher à développer un tel outil à partir de zéro. Il vaut donc mieux se tourner vers la l'extension d'une solution Open Source : prendre un designer existant et l'étendre pour inclure la possibilité de description sémantique.

#### II.2. Critères de choix

Notre objectif étant d'étendre un logiciel libre de modélisation de processus d'affaires afin qu'il puisse annoter un workflow sémantiquement. Il existe un grand nombre de logiciels libres qui permettent de modéliser les processus d'affaires. Cependant, chacun a été développé pour répondre à des besoins bien précis. Afin de faciliter l'intégration du logiciel libre choisi avec notre besoin, il est nécessaire d'établir une liste de caractéristiques souhaitées. Le tableau ci-dessous présente la liste des critères de sélection que nous allons utiliser pour choisir un logiciel pour notre étude.

Critère	Description
Un logiciel libre	Le logiciel choisi doit être un logiciel libre géré par un type de licence permettant de modifier le code source sans aucune restriction.
Code source disponible (open source)	Il ne doit pas y avoir de restrictions sur l'accès au code source.
Riche en langages supportés de modélisation	Le logiciel doit permettre la modélisation de processus d'affaires avec des langages qui supporte BPMN et XPD.L.
Extensible	Il est adapté aux extensions et modification ajoutées.

Tab 4.1: Tableau de critères de sélection de logiciels libres.

### II.2. logiciels

Nous avons déjà cité dans le chapitre 1 de ce mémoire quelques designers de workflows open source. Mais dans cette section, on va les présenter d'une manière plus détaillée pour les évaluer comparativement et choisir le meilleur qui répond à nos besoins.

#### II.2.1. Activiti

La composante de base du cadre Activiti est le moteur de processus (process engine). Ce moteur de processus fournit les capacités de base pour exécuter les modèles de processus métiers en notation BPMN 2.0 et créer de nouvelles tâches de Workflow.

Les différents composants d'Activiti sont :

- **modeler** pour la création graphique d'un workflow basé sur une architecture Web.
- **designer** un plug-in Eclipse pour le développement d'un workflow.
- **engine** le processeur de Workflow de base.
- **explorer** un outil Web pour déployer des définitions de processus.
- **rest** une application Web pour la collaboration entre les utilisateurs professionnels et les ingénieurs logiciels.

La figure suivante donne un aperçu de la pile d'outil dans Activiti.

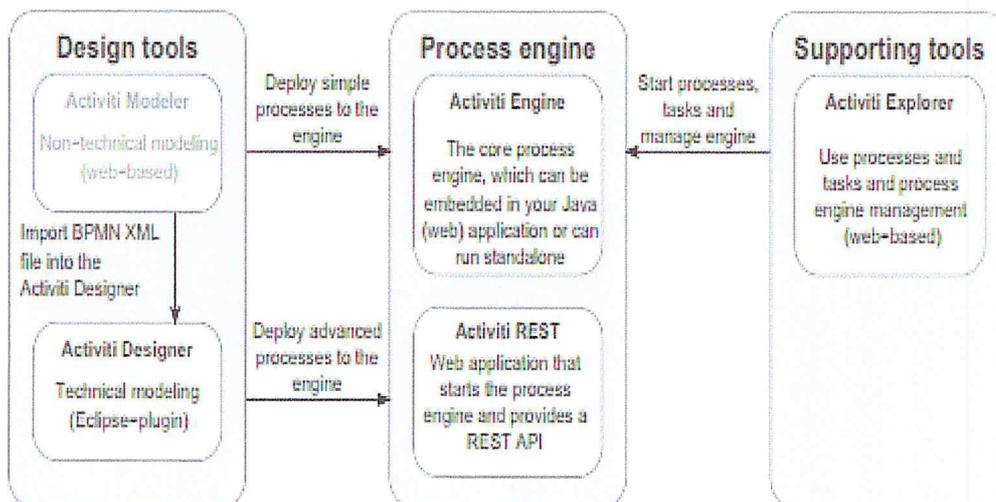


Figure 4.1: Un aperçu de la pile d'outils dans Activiti. (Rademakers, s.d.)

#### II.2.2. Signavio Core Components

BPMN 2.0 est une norme passionnante pour la modélisation des processus d'affaires et de l'exécution. Les diagrammes de processus que vous créez graphiquement peuvent être chargés dans les moteurs de processus - grâce au format d'échange XML normalisée. Les composants principaux Signavio permettent la création de base et l'échange de BPMN 2.0 diagrammes. Il ya un backend basé sur un fichier léger inclus dans le référentiel.

Le projet Signavio éléments de base contient les bases techniques de diagrammes de processus d'édition.

Mais l'inconvénient de cet éditeur c'est que ne supporte pas beaucoup de langage comme le XPDL par exemple d'après ce qu'on a testé lors de l'installation.

#### II.2.4. Oryx editor

Oryx est un projet académique Open Source qui permet d'ajouter des prototypes à une infrastructure puissante de modélisation de processus. Il est devenu une plateforme ouverte pour les développements concernant la modélisation des processus métiers.

Oryx possède deux points forts, le premier est sa capacité supporter n'importe quel langage de modélisation de processus d'affaires grâce au module de définition de langage Stencil Set et son infrastructure de plugiciels qui lui permet d'être ouvert facilement à de nouvelles fonctionnalités, le deuxième est que les utilisateurs peuvent créer des modèles de processus d'affaires dans un simple navigateur et de les partager avec tous les autres utilisateurs. (FODIL, 2011)

#### Avantages d'Oryx :

- ✓ Un outil puissant pour la modélisation de processus graphique.
- ✓ Oryx s'exécute dans un navigateur web, aucune installation n'est requise.
- ✓ Oryx est une plate-forme de processus - les parties prenantes peuvent accéder à des modèles de processus via le Web.
- ✓ Basé sur la technologie Web, son intégration dans l'IT de l'entreprise est simplifiée.
- ✓ Oryx est extensible, de nouvelles fonctions peuvent être ajoutées via un mécanisme de plug-in.
- ✓ Oryx supporte directement BPMN, EPC, et d'autres langages de modélisation de processus.
- ✓ Authentification des utilisateurs via OpenID.
- ✓ Oryx est une plate-forme de processus - les parties prenantes peuvent accéder à des modèles de processus via le Web.
- ✓ Oryx est basé sur des normes Internet ouvertes.

#### Avantages pour les développeurs :

- ✓ Code côté client écrit en Javascript, code côté serveur écrit en Java.
- ✓ Facile: définition déclarative des langages nouveaux de modélisation de processus (nous les appelons stencil set).
- ✓ Extension facile des fonctionnalités de l'éditeur via un mécanisme de plug-in (petit noyau, presque toutes les caractéristiques sont mises en œuvre sous forme de plugins).
- ✓ La technologie de rendu utilisé est SVG et XHTML.

#### II.3. Le choix final

Notre choix s'est porté sur l'outil "Oryx", car ce dernier correspond à nos exigences répondant à tous nos besoins concernant la possibilité d'extensibilité, disponibilité du code source et surtout la richesse du langage de modélisation, sachant qu'il est le seul parmi la liste proposée ci-dessus qui admet le plugin d'exportation et l'importation en XPDL. Ceci a été constaté par nos soins propres après installation et test de chaque outil. Oryx est donc le logiciel le plus approprié au contexte de notre travail par le fait qu'il puisse être personnalisé, ce qui nous permettra d'introduire notre annotation sémantique, comme le montre le tableau ci-dessous.

Critère Logiciel	Logiciel Libre	Disponibilité du code Source disponible	modélisation supportés Richesse en langages de	Extensible
Signavio	+	+	-	-
Activiti	+	+	+	-
Oryx	+	+	+	+

Tableau 4.2: Comparaison entre les différents logiciels de modélisation de workflows.

### III. Langages de modélisations et d'exécution utilisés

Comme cité précédemment, dans le chapitre qui concerne les processus métiers et workflow; il existe beaucoup de langages de modélisations comme (UML, EPC, Petri Net, BPMN..etc.) ainsi que divers langages d'exécution tel que ( XPDL, BPEL..etc.).

Dans la section suivante, on va cité les raisons de choisir les langages utilisés dans notre projet.

#### III.1. Langage de modélisation choisi

Aujourd'hui, La spécification BPMN est devenue stable et elle est adaptée aux besoins du marché où plusieurs éditeurs l'ont adoptée et introduite dans leurs outils. Cette spécification hérite de tant d'expérience et émane de telles fondations qu'elle offre une puissance d'expression infinie : elle s'applique à la fois aux processus privés (internes à une organisation ou à une entreprise) ou collaboratifs (collaborations entre deux organisations ou deux entreprises) et c'est qui est très important pour notre cas. Aussi elle permet la traduction automatique d'un processus modélisé graphiquement dans un format compréhensible par un moteur d'exécution de processus .

## ***Chapitre 05***

# ***Description et mise en œuvre de la solution proposée***

#### I. Introduction

Dans ce chapitre, on va présenter la problématique majeure de la description sémantique des Workflows, ainsi que notre description du solution proposée, qui illustre les nouvelles fonctionnalités apportées par notre procédé d'annotations sémantique à Oryx editor.

Par la suite on va se concentré sur la mise en œuvre de l'outil d'annotation en implémentant les résultats de la recherche bibliographique et les méthodes choisies dans les chapitres précédentes.

#### II. Limites des technologies Workflow

Certes, l'existence de l'approche BPM, gère l'exécution des processus entre les entreprises et unifie leurs traitements, dans le but de fournir les outils et techniques nécessaires pour la modélisation des processus, en mettant l'accent sur la représentation graphique des processus d'affaires, où cette représentation est décrite par une description syntaxique comme XPDL, qui permet de définir un processus métier à l'aide du langage XML.

Mais cette approche n'apporte pas un plus en terme d'explicitation des termes et concepts manipulés, et donc le problème d'exploitation des informations réside toujours. Devant cette problématique il est impératif d'ajouter la notion de sémantique, qui va pouvoir améliorer l'interprétation des données, ce qui va permettre une meilleure organisation des informations et une facilité d'accès afin de satisfaire le besoin majeur recherché par toutes les entreprises, qui est l'automatisation des tâches.

#### III. Description du travail réalisé

Comme il a été expliqué précédemment, le but de notre projet est de mettre en place un outil de modélisation et de description sémantique des processus métiers pour entreprise.

Nous avons optés pour l'extension d'un outil Open Source existant, pour y inclure des fonctionnalités de description sémantique. Pour cela, nous avons décidé d'utiliser Oryx editor en raison des arguments avancés dans le chapitre précédent. L'extension de ce dernier se ferait grâce à un mécanisme de plug-in, qui est une possibilité d'extension offerte dans Oryx, qui permettra d'annoter les propriétés des concepts servant à la modélisation des workflows.

Pour la réalisation de notre solution, nous avons défini un nouveau Stencil-Set basé essentiellement sur le Stencil-Set<sup>4</sup> de BPMN 1.2 offert par défaut dans Oryx. Ce nouveau Stencil-Set permet la prise en charge de l'annotation sémantique des Workflows BPMN 1.2 selon le procédé "Open Annotation".

De plus, une nouvelle fonctionnalité a été rajoutée à Oryx pour assurer la conversion d'un modèle BPMN 1.2 incluant des annotations sémantiques vers un format XML basé sur l'ontologie "OA4XPDL" que nous avons défini par extension de l'ontologie OA du modèle "Open Annotation".

Pour pouvoir réaliser la solution proposée ci-dessus, il faut d'abord bien connaître l'architecture et le fonctionnement d'Oryx.

---

<sup>4</sup> Stencil-Set: un ensemble de nouveaux symboles à utiliser dans la modélisation des processus métiers.

#### IV. Modélisation des nouvelles fonctionnalités fournies par notre procédé

Comme nous l'avons vu, Oryx est un designer non seulement open-source mais aussi extensible; et cette dernière propriété est très importante car elle nous permet d'ajouter de nouvelles fonctionnalités pour répondre à la problématique présentée ci-dessus et ainsi implémenter la solution proposée.

Le schéma suivant représente les deux gros nouvelles fonctionnalités et approches apportées dans l'outil utilisé.



Figure 5.2: Les fonctionnalités ajoutées dans oryx editor.

on va présenter dans le schéma ci-dessous la démarche utilisée, les intégrations réalisées et l'ensemble des tâches pour aboutir les deux fonctionnalités représentées ci-dessus; et donc le résultat voulu qui est l'annotation sémantique des Workflows inter-organisationnels basé sur le Framework "Open Annotation".

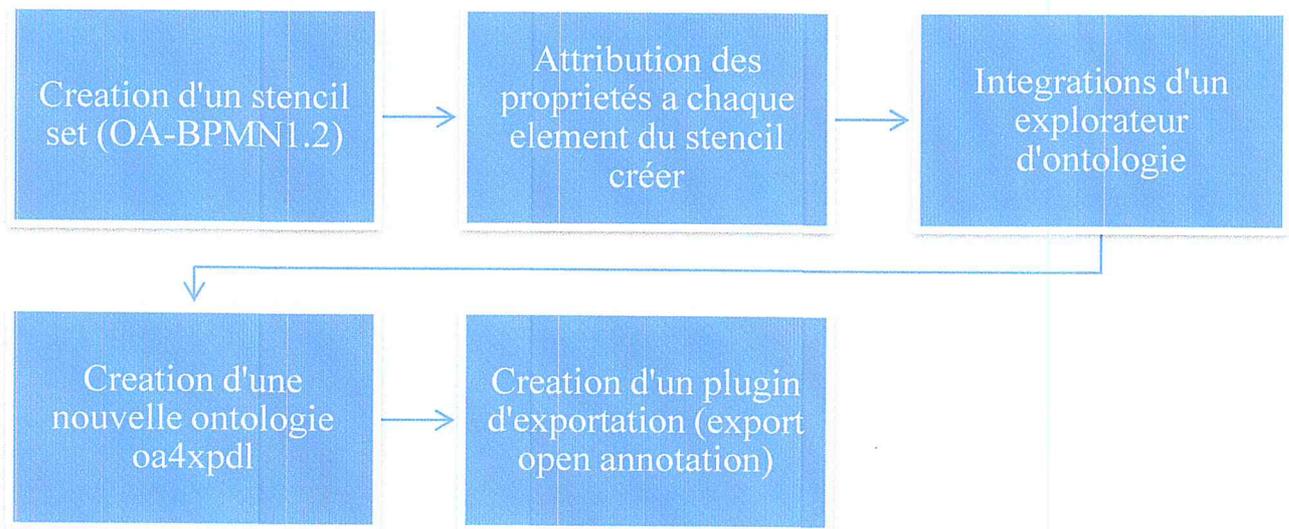


Figure 5.3: La démarche utilisée pour annoter sémantiquement les Workflows inter-organisationnels.

#### ➤ descriptions détaillées

Le diagramme représenté ci-dessus illustre les nouvelles fonctionnalités à ajouter dans Oryx tel que nous allons présenter d'une manière assez détaillée comme se suit :

**1. Annotation des Workflows inter-organisationnels :** elle représente le cœur de notre travail car elle résout le problème associé à l'interopérabilité des Workflows existants et elle enrichit la représentation graphique et la description syntaxique du Workflow modélisé. Cette fonctionnalité admet:

- a) **Création d'un nouvel Stencil Set (OA-BPMN1.2):** En générale un Stencil set est un module de définition des langages de modélisation. Dans notre cas on a ajouté un nouveau langage d'annotation basé sur le "Open Annotation" tel qu'on a spécifié l'ensemble des symboles utilisés par le langage et les règles qui gèrent leur utilisation.

Les symboles utilisés sont : Annotation, Body et Target. Ces trois éléments sont des éléments de modèle de base d'Open Annotation où:

**Annotation:** a un seul Body et une Target.

**Body:** qui est soit un commentaire soit une autre ressource descriptive.

**Target:** que le body est en quelque sorte à propos.

Ils ont représentés sous la forme des ellipse et la Target sous forme d'une flèche comme indique le modèle de base d'Open Annotation dans le chapitre 2 de ce mémoire.

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

b) **Attribution des propriétés a chaque élément du stencil créer:** Chaque élément (Annotation, Body et Target) doit avoir un ensemble de propriétés nécessaires pour qu'il soit fonctionnel, utile et enrichir l'annotation.  
par exemple, Annotation est caractérisé par (id, name, has body où on doit affecté le body associer ainsi has Target une motivation fixe qui est Semantique\_Annotation).  
pour le Body ( id, name, HasUri où on doit affecté le concept tiré d'une ontologie à l'aide de l'explorateur intégré).  
finalement le Target par (id, name, ConformeTo a la valeur hasXpdlSelector, Type qui est fixe par XpdlSector, HasSource represente le Workflow annoté et hasValue qui représente l'élément annote dans le Workflow ).

c) **Intégrations d'un explorateur d'ontologie:** L'annotation sémantique des Workflows inter-organisationnel sert a tiré un concept associé (URI) d'une ontologie; c'est pourquoi on a intégré un explorateur d'ontologie. Ce dernier contient un ensemble des ontologies dédié a l'annotation sémantique des Workflow inter-organisationnels d'une façon assez simple qu'on va la présenté dans la partie de l'implémentation.

2. **Exportation des Workflows inter-organisationnel annotés:** après avoir annoté les workflows, il est nécessaire maintenant de sauvegarder ces annotations pour les réutiliser plus tard. Oryx editor utilise plusieurs mécanisme d'exportations mais il n'inclut pas l'exportation de ces annotations, chose qu'on a rajouté comme élément nouveau dans notre travail.

a) **Création d'une nouvelle ontologie oa4xpdl:** Pour que la syntaxe des fichiers générés soit conforme au modèle Open Annotation et qu'elle prenne aussi en compte les spécificités des langages de workflows BPMN/XPDL, nous avons construit une extension de l'ontologie OA (Open Annotation), qui est une nouvelle ontologie appelée OA4XPDL.

b) **Création d'un plugin d'exportation (export open annotation):** Comme nous avons utilisés le Workflow designer "Oryx" ce dernier ne supporte pas le nouveau module d'annotation des Workflows créer en BPMN1.2 intégré, par l'exportation qui se fait par le biais du plugin dans sa version originale qui supporte XPDL comme format de sauvegarde. Donc, nous avons dû créer un nouveau plugin (export open annotation), qui exporte les annotations créées (le stencil set ajoutés) sous forme d'un fichier XML.

Dans la première partie de ce chapitre nous avons présenté une description de la solution proposée, l'étape suivante sera d'en réaliser l'implémentation.

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

Dans le but d'identifier les endroits où on peut apporter les changements nécessaires pour l'introduction de notre procédé d'annotation sémantique, au sein d'Oryx editor, nous allons analyser le code source d'Oryx, il faut dire que la documentation y relative n'est pas disponible.

Avant d'entamer l'implémentation, il faudra d'abord comprendre la structuration du projet Oryx, afin de réaliser la phase de modification. Nous verrons aussi l'ensemble des langages et des outils nécessaires, pour la réalisation de la solution proposée et nous terminerons par des aperçus sur le résultat final.

#### V. Eléments du Workflow à annoter

Chaque workflow est composé de plusieurs éléments pour présenter un processus métier, ces éléments sont par exemple : les activités, les événements, les messages, les dataobjects, etc.

Dans un premier temps, il a été convenu que l'annotation sémantique serait restreinte aux seuls concepts d'« activité », « data object » et « message flow ».

Le choix de ces trois éléments est fait pour les raisons suivantes :

- i. Activité représente un élément indispensable dans un workflow car elle est caractérisé par un nom qui peut différer d'une entreprise à l'autre, ce qui implique l'association d'une description sémantique.
- ii. DataObject Sont des objets utilisés pour représenter les données nécessaires pour l'exécution d'une activité, ainsi que celles produites par des activités.
- iii. Message Flow C'est à ce niveau que les informations et les données circulent entre les Workflows partenaires, ce qui nécessite de les annoter pour leur attribuer un sens sémantique explicite.

En résumé, le plug-in devra donc rajouter les propriétés d'annotation nécessaires aux modèles et permettre à l'utilisateur de les annoter en sélectionnant le concept approprié parmi ceux de l'une des ontologies proposées. A cela, s'ajoute le fait que Oryx permet de sauvegarder les modèles de processus sous divers formats, et qu'il faudra donc modifier les mécanismes de générations de ces formats de sauvegarde, pour y inclure les descriptions sémantiques.

En conclusion, sous sa forme finale notre solution :

- Permettra d'annoté les « activités (Task) » d'un processus ;
- Permettra d'annoté les «Data Object » d'un processus ;
- Permettra d'annoté les « Message Flow » d'un processus ;
- De générer les propriétés de description sémantiques dans des fichiers appropriés respectant la syntaxe définie d'Open Annotation, en utilisant un nouveau plugin créer pour l'export des annotations.

#### VI. Architecture et fonctionnement du plugin ORYX

Comme nous l'avons indiqué dans le chapitre précédent, Oryx possède deux points forts, le premier est sa capacité de supporter n'importe quel langage de modélisation de processus métiers grâce au module de définition de langage (Stencil-Set) et de son infrastructure de Plugiciel (basée sur les plugins clients et les plugins serveurs), ce qui lui permet d'être ouvert facilement à de nouvelles fonctionnalités, le deuxième avantage est que les utilisateurs peuvent créer des modèles de processus métiers et les partager.

Les plugins d'Oryx se présentent, en général, sous la forme suivante :

- Un Plug-in côté serveur : constitué de Servlets Java, généralement destinés à réaliser les gros traitements. Leurs URLs doivent être rajoutées dans les fichiers de configuration « config.js » et « web.xml ».
- Un plug-in côté client : Il représente l'interface utilisateur pour l'accès à la nouvelle fonctionnalité. Il définit une façade pour l'accès au plug-in Serveur correspondant, ainsi que pour gérer les événements pour la communication avec les autres plug-ins. Cette partie est développée en JavaScript et utilise l'API « Facade » fourni par Oryx, pour accéder aux différentes fonctionnalités offertes par le Core Oryx.

Pour que le plugin soit fonctionnel il doit être ajouté à un registre des plugins et aux profils des utilisateurs qui doivent l'utiliser. Le registre des plugins est un fichier XML (plugins.xml) qui décrit tous les plugins disponibles sur l'application, et le fichier « profils.xml » associe à chaque profil l'ensemble des plugins dont l'utilisateur de ce profil a l'accès. Le plug-in client assure aussi la conversion des modèles de processus au format JSON pour la communication avec la partie serveur d'Oryx.

- Eventuellement, un nouveau « Stencil-Set » : un ensemble de nouveaux symboles à utiliser dans la modélisation des processus métiers.

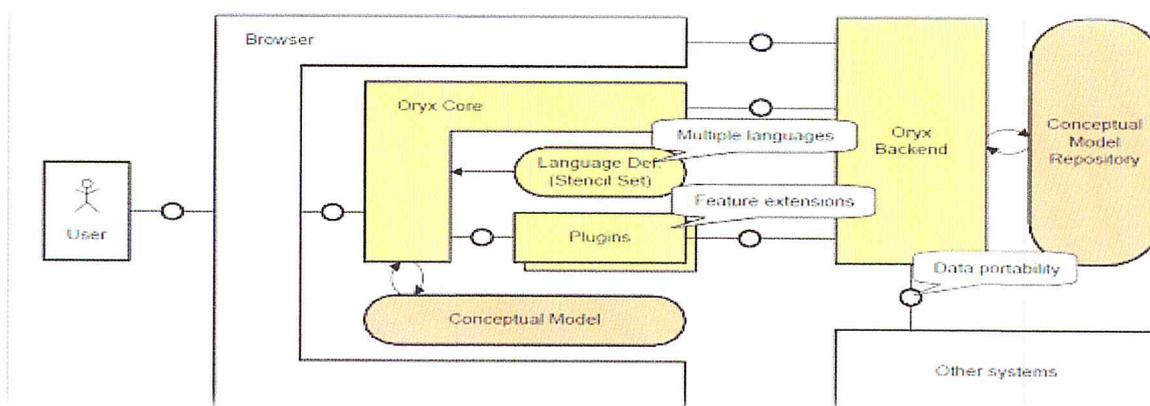


Figure 5.1: Architecture générale d'Oryx.(Fodil,2011)

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

Comme montré, la Figure ci-dessous représente l'architecture générale de Oryx editor, elle regroupe plusieurs composants. Pour notre cas on s'intéresse beaucoup plus sur deux principaux modules.

Le premier de ces modules est concernant les langages de stencil set où on va étendre notre nouveau Stencil set d'annotation sémantique en le nommant "OA-BPMN1.2" (indiqué par num1), et par la suite on va créer un nouveau plugin qui sert à l'exportation de différents annotations; Ce plugin va être intégré dans le module des plugin d'Oryx (indiqué par 2).

Pour mieux illustrer les composants concernés, voir la figure suivante.

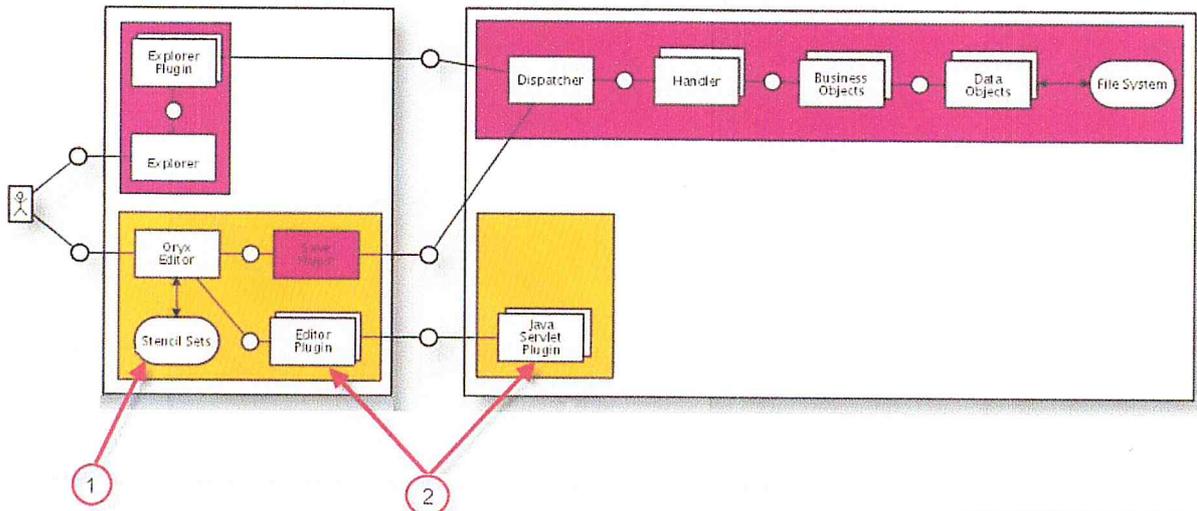


Figure 5.2: Composants de projet Oryx (Peters, 2009)

## VII. Structuration du projet Oryx

Le détail des étapes de configuration pour l'environnement de développement d'Oryx est décrit dans l'Annexe D.

La dernière version du code source d'Oryx est disponible sur le lien suivant "<https://code.google.com/p/oryx-editor/source/checkout>".

Oryx est une application client-serveur, la partie cliente est développée en JavaScript alors que la partie serveur est écrite en Java. On verra dans ce qui suit quelque composants essentiels du logiciel Oryx.

Oryx est constitué des composants suivants :

### a. Oryx editor :

C'est une interface *cliente* destinée aux utilisateurs, accessible à partir de n'importe quel navigateur Internet, pour modéliser des workflows en utilisant différents formalismes.

### b. Stencil Set :

C'est un module de définition des langages de modélisation, qui permet à Oryx de supporter différents langages de modélisation de processus d'affaires.

Donc pour l'ajout d'un nouveau langage de modélisation, il faut juste spécifier l'ensemble des symboles utilisés par le langage et les règles qui gèrent leur utilisation.

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

Oryx utilise trois technologies dans la définition de langages de modélisation de processus d'affaires :

- La notation *JSON*<sup>5</sup> qui sert à décrire les langages de modélisation, leurs propriétés et les règles qui manipulent les symboles formant les langages.
- *SVG*<sup>6</sup> (*ScalableVectorGraphic*) pour décrire les représentations graphiques des symboles.
- *JavaScript* pour définir la partie dynamique des langages de modélisation en décrivant le comportement des objets qui composent les langages de modélisation.

#### c. Les Plugins :

Grâce aux plugins on peut ajouter des fonctionnalités au navigateur selon le besoin, car les plugins regroupent la majorité des fonctionnalités d'Oryx.

#### d. Système principal d'Oryx :

C'est le cœur d'Oryx, et il est implanté sur un serveur Web capable d'exécuter le code java (comme Apache TomCat ou GlassFish par exemple). Il est composé de l'implantation du code.

## VIII. Langages et environnement de développement

### VIII.1. Les langages de programmation utilisés

Dans ce qui suit on fera une petite présentation des langages qu'on a utilisé afin d'introduire notre annotation dans Oryx .

On a principalement utilisé deux langages de programmation, il s'agit du langage Java du côté serveur, et du langage JavaScript du côté client.



#### ➤ JAVA

Java est un langage de programmation et une plate-forme informatique qui ont été créés par Sun Microsystems en 1995. Les applications Java peuvent être exécutées sur tous les systèmes d'exploitation pour lesquels a été développée une plate-forme Java, dont le nom technique est JRE (Java Runtime Environment - Environnement d'exécution Java). Cette dernière est constituée d'une JVM (Java Virtual Machine - Machine Virtuelle Java), le programme qui interprète le code Java et le convertit en code natif. Mais le JRE est surtout constitué d'une bibliothèque standard à partir de laquelle doivent être développés tous les programmes en Java. C'est la garantie de portabilité qui a fait la réussite de Java

<sup>5</sup> JSON: JavaScript Object Notation, est un format de données textuelles dérivé de la notation des objets du langage JavaScript.

<sup>6</sup> SVG: est un [format de données](#) conçu pour décrire des ensembles de [graphiques vectoriels](#) et basé sur [XML](#).

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

dans les architectures client-serveur en facilitant la migration entre serveurs, très difficile pour les gros systèmes.

## JavaScript

### ➤ JavaScript

JavaScript est un langage de programmation de scripts principalement utilisé pour les pages Web interactives. C'est un langage orienté objets à prototype, c'est à dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas instanciés au sein de classes mais qui sont chacun équipés de constructeurs permettant de générer leurs propriétés, et notamment une propriété de prototypage qui permet d'en générer des objets héritiers personnalisés.

Aujourd'hui JavaScript est maintenant la propriété d'Oracle.

### ➤ XML

C'est un langage informatique de balisage extensible qui dérive du SGML. Il est dit extensible car sa syntaxe permet de définir différents espaces de noms, c'est-à-dire des langages avec chacun leur vocabulaire et leur grammaire, comme XHTML, XSLT, SVG... XML est reconnaissable par son usage des chevrons (< >) encadrant les balises. L'objectif initial est de faciliter l'échange automatisé de contenus complexes (arbres, texte riche...) entre systèmes d'informations hétérogènes (interopérabilité).

Les langages associés à une application XML respecte généralement certains principes :

- La structure d'un document XML est définie et validable par un schéma,
- Un document XML est entièrement transformable dans un autre document XML.

## VIII.2. Environnement de développement

### ➤ Serveur Web Apache Tomcat

Tomcat est un serveur d'application de l'Apache Software Foundation qui exécute servlets Java et rend les pages Web qui incluent des Java Server page de codage. Tomcat est le résultat d'une collaboration ouverte de développeurs et est disponible sur le site Web Apache dans les deux versions binaires et sources. Tomcat peut être utilisé comme un produit autonome avec son propre serveur Web interne ou avec d'autres serveurs Web, y compris Apache, Netscape Enterprise Server, Microsoft Internet Information Server (IIS) et Microsoft Personal Web Server. Tomcat nécessite Java Runtime un environnement d'entreprise qui se conforme à JRE 1.1 ou ultérieure.

### ➤ Eclipse

Développer à l'origine par I.B.M, Eclipse est un environnement de développement intégré (Integrated Development Environment) dont le but est de fournir une plate-forme modulaire pour permettre de réaliser des développements informatiques. Son plus grand

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

avantage c'est qu'il représente une plate-forme ouverte pour le développement d'applications et extensible grâce à un mécanisme de plug-in.

#### IX. Intégration de l'annotation sémantique dans Oryx editor

Afin de personnaliser Oryx pour qu'il puisse intégrer notre module d'annotation, on a procédé à quelques modifications du code source.

La partie dans laquelle on va apporter le plus de modifications sera la partie **Gestion de l'éditeur graphique**, car elle comporte les programmes nécessaires à la spécification des modules de définition de langage (Stencil Set) et à la représentation des processus d'affaires dans l'interface graphique d'Oryx. Cette partie a été développée avec Java.

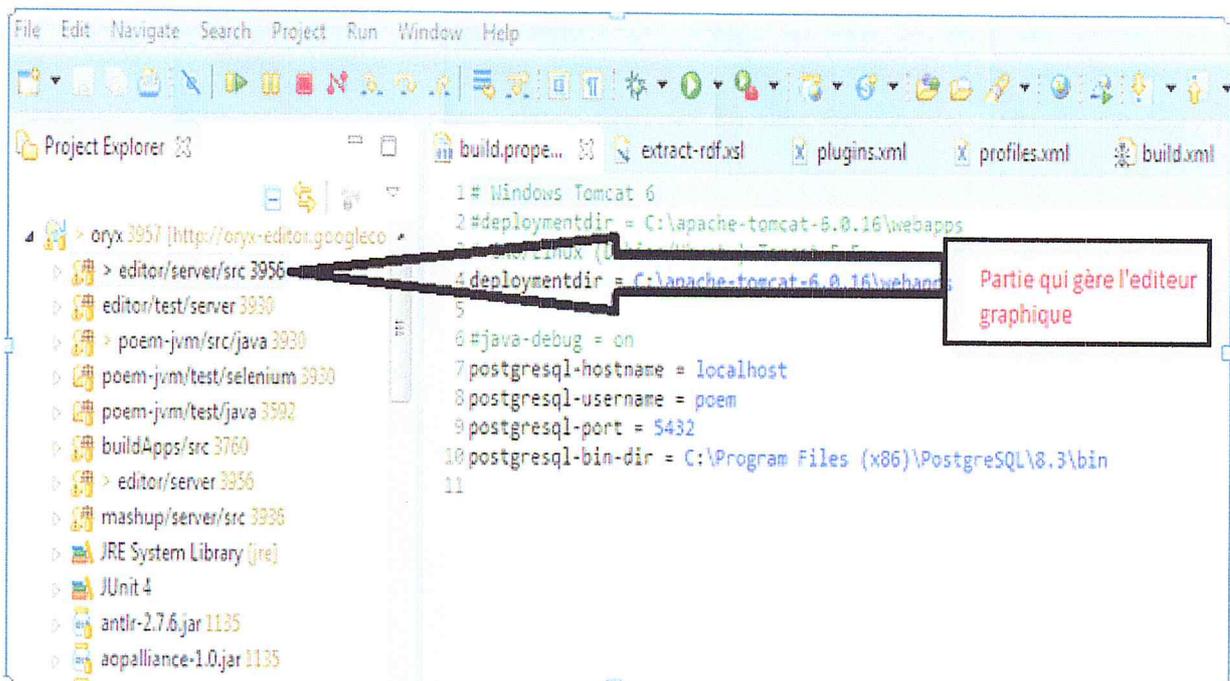


Figure 5.4: Structure du code source d'Oryx.

#### Modification JavaScript :

Cette modification touche le côté client du logiciel et permet à l'utilisateur de pouvoir intégrer **graphiquement** ses annotations. Dans cette partie les modifications sont faites dans deux endroits qui sont :

1. Les fichiers **.json** qui représente le code source contenant la définition des **Stencil Set** qui représentent les propriétés et les règles du langage de modélisation, On a ajouté les éléments nécessaires pour l'annotation sémantique qui sont les trois éléments (Annotation, Body et Target) tel que en spécifiant pour chacun d'eux leurs attributs ainsi leurs règles pour facilite la tâche d'annotation.
2. Les fichiers **.js** : notre intervention consiste à introduire le code source qui sert à envoyer la définition des objets manipulés graphiquement vers le serveur. Dans notre cas on a crée un bouton dans la liste d'exportation qui sert à lancer la requête de transformation de graphe d'annotation vers un fichier Open Annotation annoté.

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

#### Modification XML :

Dans cette partie on a procédé aux changements nécessaires qui servent à valider nos fichiers annotés, parmi ces changements on a introduit la définition des stencils rajoutés dans les fichiers `.xsd` qui contiennent la définition des éléments du workflow.

On a aussi injecté les modifications nécessaires dans les fichiers « `.xslt` » qui couvre la transformation et l'interprétation de la modélisation graphique du workflow modélisé vers celle de Open Annotation.

#### Modification Java :

Le but de notre modification est de permettre à l'utilisateur de définir l'annotation des éléments BPMN1.2 et ensuite voir son annotation traduite en XML, chose qui nécessite la création de nouvelles classes java qui assurent cette transformation. Pour cela, on a créé des classes contenant l'ensemble des propriétés, méthodes et rôles correspondant à l'ensemble des éléments de stencil ajouté, soit Annotation, Body ou Target ainsi que d'autres classes qui gèrent la transformation et la construction du fichier annoté.

### X. La réalisation de la solution

Dans ce qui suit, nous allons présenter notre nouvelle version d'Oryx avec un nouveau mécanisme d'annotation "Open Annotation", pour le langage des description syntaxique de Workflows BPMN 1.2, ainsi que les étapes suivies pour son développement, sa mise en œuvre et finalement son test.

#### X.1. Aperçu original de Oryx editor

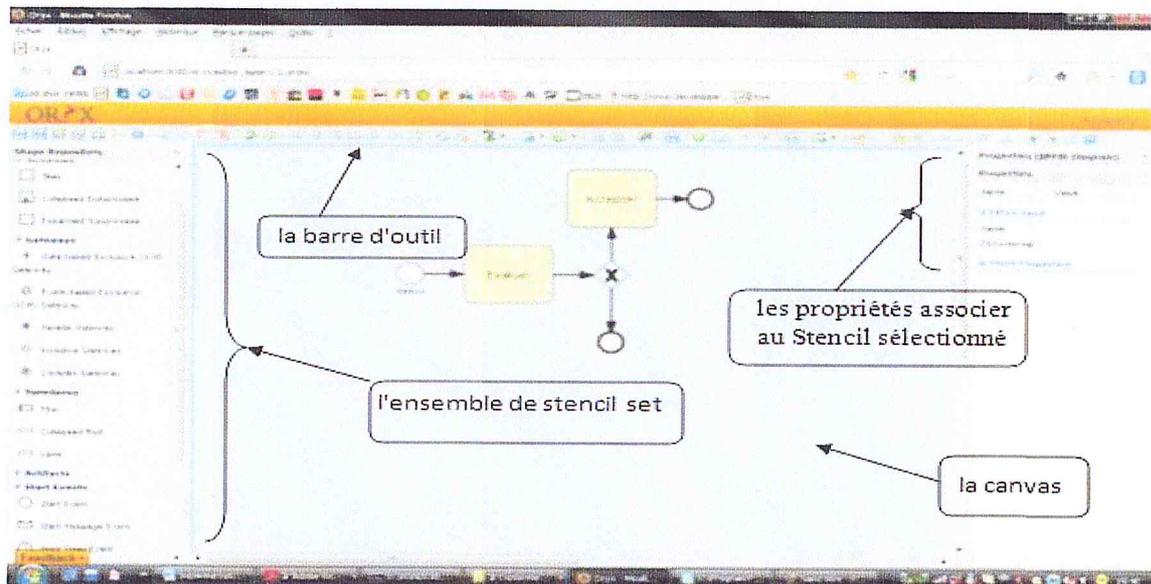


Figure 5.6 : Un Aperçu originale de oryx editor.

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

#### X.2. Création d'un nouveau stencil set pour "OA-BPMN 1.2"

Comme on a indiqué ci-dessus, Oryx contient un ensemble de stencil set qui servent à définir les langages de modélisation. Nous avons rajouté un nouveau stencil set pour supporter le nouveau module d'annotation sémantique.

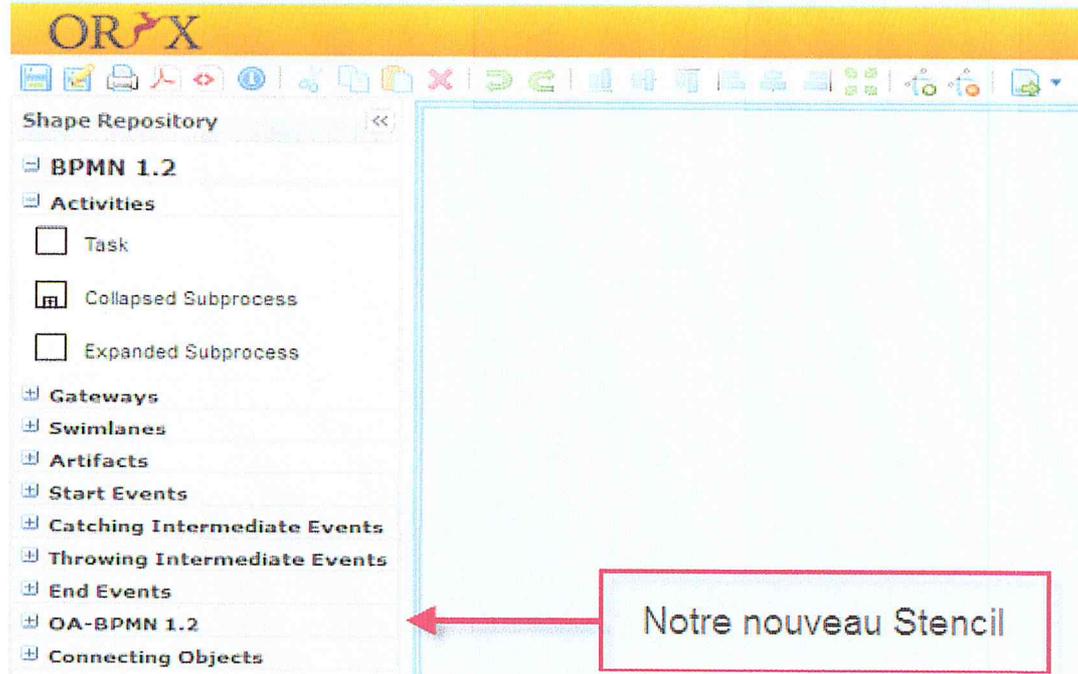


Figure 5.7: Le nouveau Stencil set "OA-BPMN 1.2"

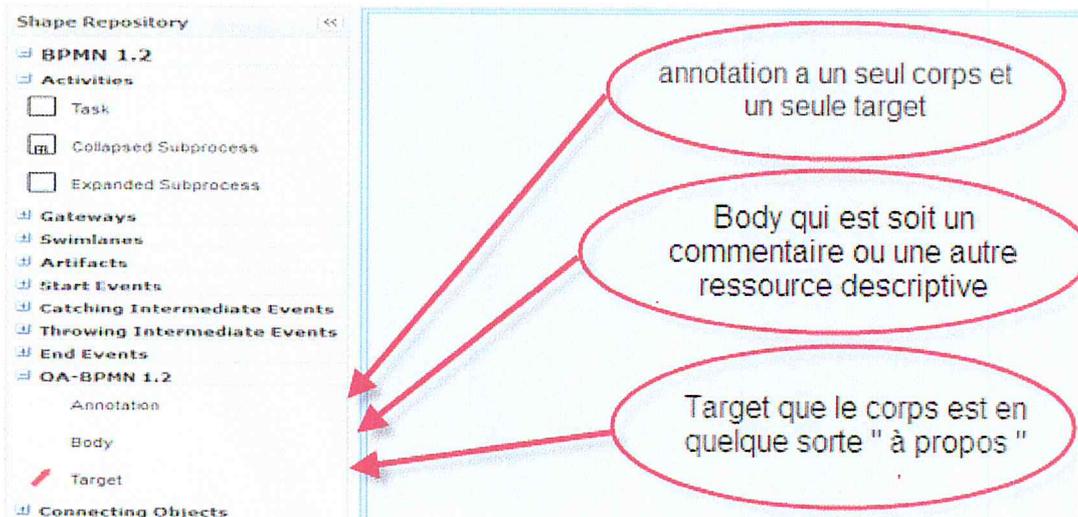


Figure 5.8: Les éléments du Stencil Set "OA-BPMN1.2"

- **les propriétés associées** : chaque élément du Stencil Set ajouté doit avoir un ensemble de propriétés, dans le but de l'identifier et de le caractériser comme montre les figures suivantes:

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

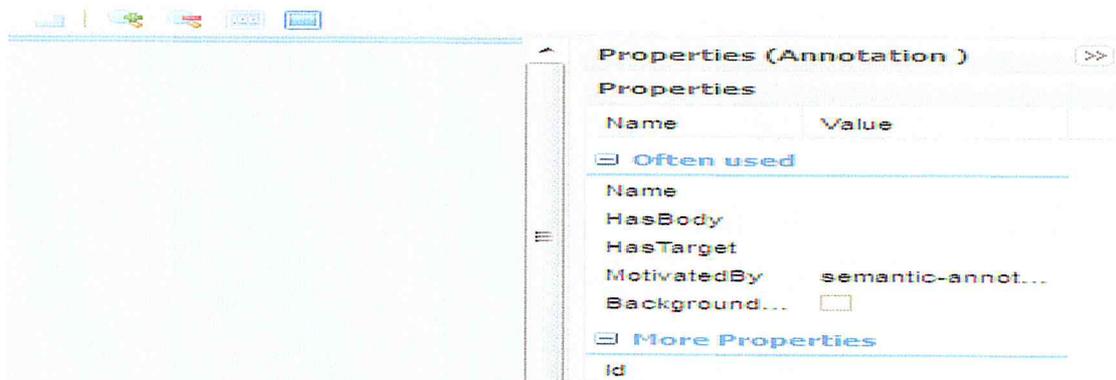


Figure 5.8 Les propriétés associées au Annotation

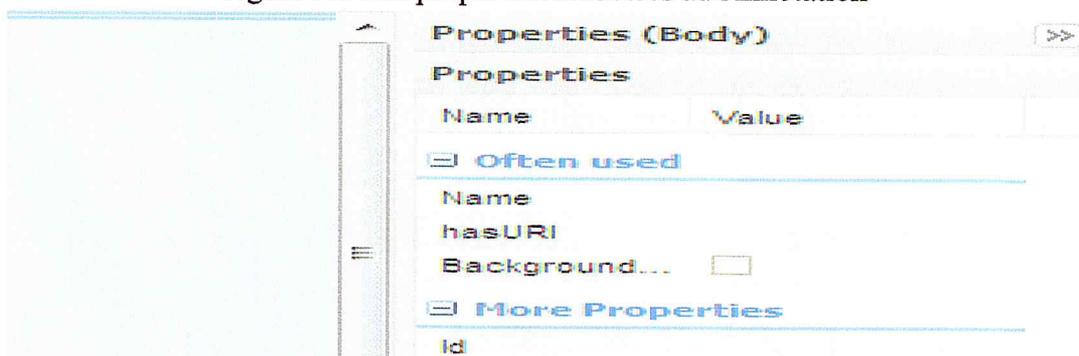


Figure 5.9: Les propriétés associées au Body

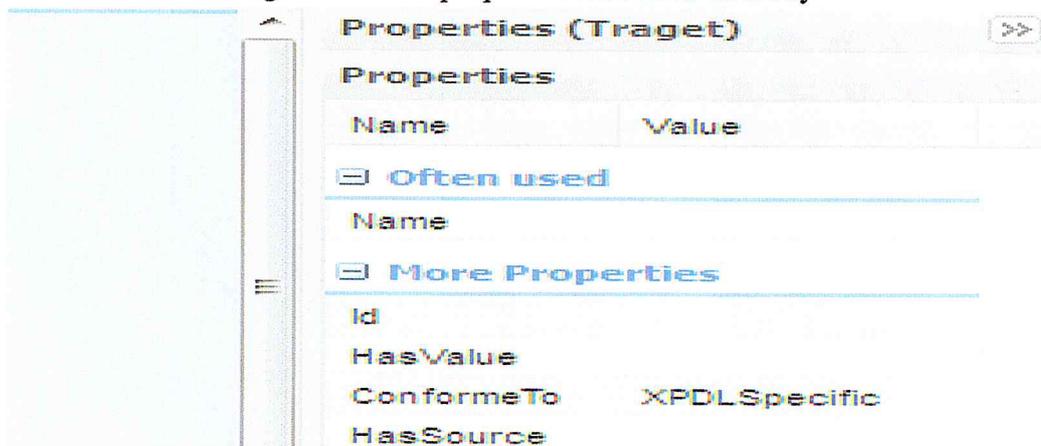
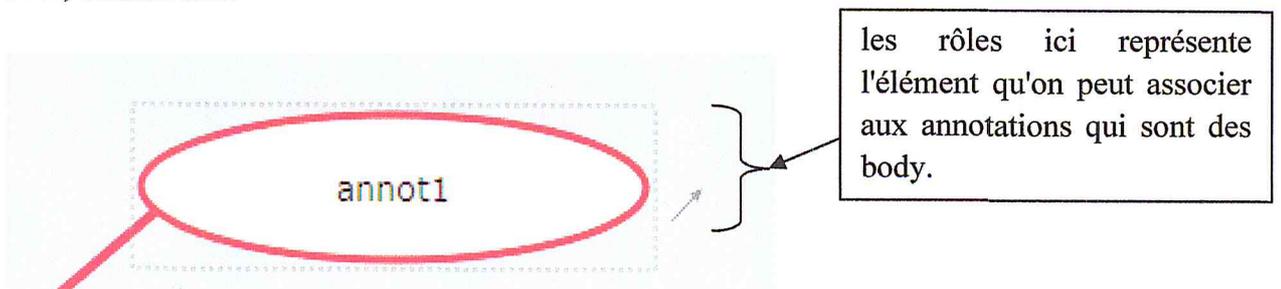


Figure 5.10: Les propriétés associées au Traget

- **Association des rôles aux stencils:** pour que l'élément ajouté soit fonctionnel dans Oryx editor, on doit lui affecter des rôles qui sert a définir l'élément qu'on peut associer avec; comme suit:



## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

Figure 5.11: Les rôles associées au Annotation

#### X.3. Le nouveau plugin d'exportation

Comme nous souhaitons annoter les Workflows créer en BPMN1.2, leur exportation doit se faire par le biais du plugin qui supporte XPDL comme format de sauvegarde, et qui dans sa version originale ne prend pas en considération les nouveaux stencils ajoutés (relatifs aux annotations). Donc, nous avons dû créer un nouveau plugin, qui exporte les annotations créées sous forme d'un fichier XML.

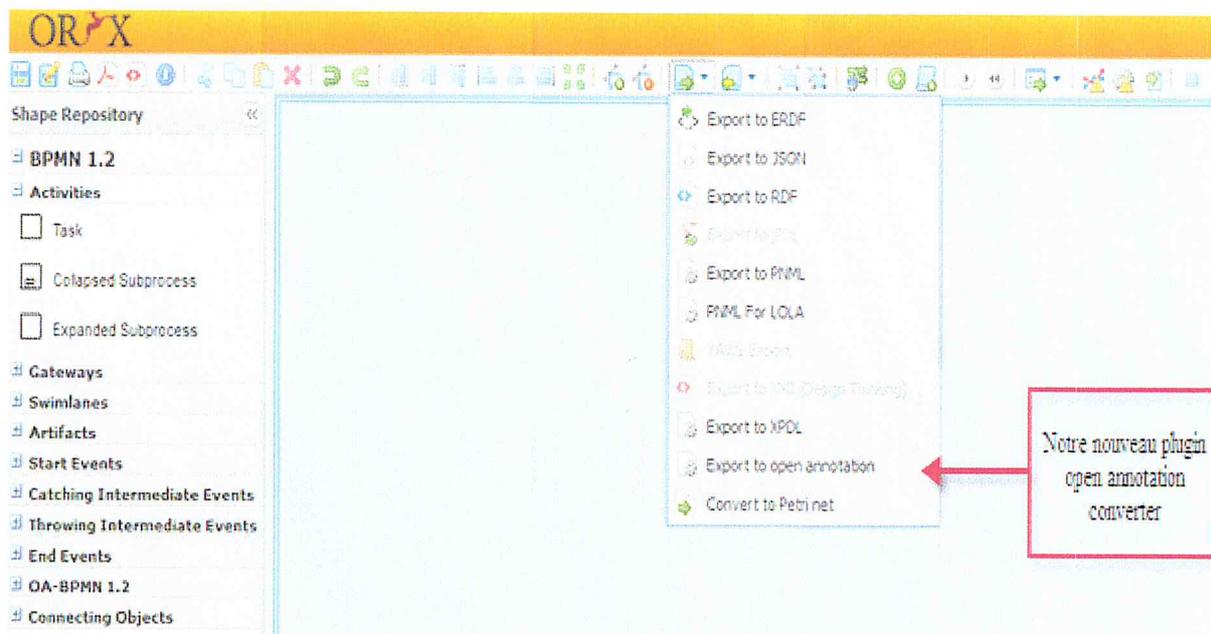


Figure 5.12: Le nouveau plugin d'exportation

#### X.4. La création d'une nouvelle ontologies OA4XPDL

Pour que la syntaxe des fichiers générés soit conforme au modèle Open Annotation et qu'elle prenne aussi en compte les spécificités des langages de workflows BPMN/XPDL, nous avons construit une extension de l'ontologie OA (Open Annotation), qui est une nouvelle ontologie appelée OA4XPDL. Cette ontologie OA4XPDL hérite tous les concepts de OA et rajoute de nouveaux attributs propres à XPDL comme les '*IDs*' des éléments, de nouveau type de motivation nommé '*semantic\_annotation*', et un nouveau type de selector appelé '*XPDLSelector*'.

Cette ontologie est représentée en détail dans l'annexe E de ce mémoire.

## XI. Démarche d'annotation sémantique d'un Workflow inter-organisationnel dans Oryx editor

### 1. Authentification

L'utilisateur doit d'abord s'identifier pour pouvoir sauvegarder et retrouver les processus qu'il modélise. Actuellement, l'éditeur graphique d'Oryx requiert l'utilisation du Navigateur Firefox pour pouvoir éditer correctement les processus.

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

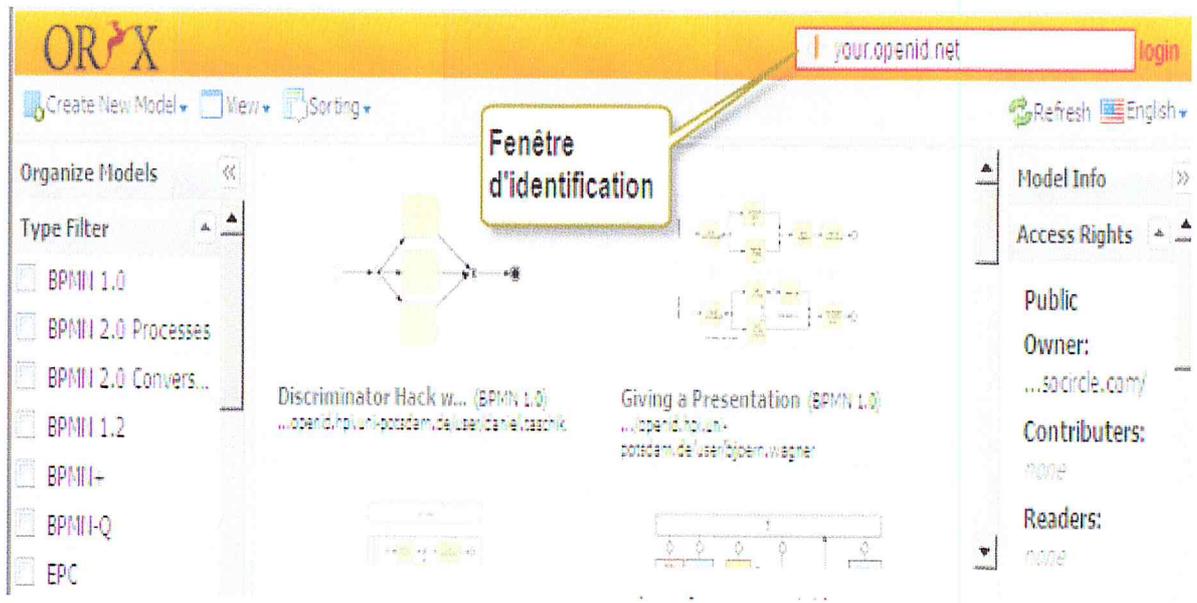


Figure 5.13: L'authentification

Oryx utilise le système d'identification personnel OpenID, pour identifier les utilisateurs. Tout nouvel utilisateur doit d'abord s'inscrire sur un serveur OpenID (comme MyOpenID ou bien Symantec), puis utiliser son code d'accès enregistré pour se connecter au serveur Oryx. A chaque accès, l'utilisateur doit entrer son UIR OpenID dans le champs de login, où fenêtre de confirmation va s'afficher pour authentifier l'utilisateur sur le service OpenID correspondant :

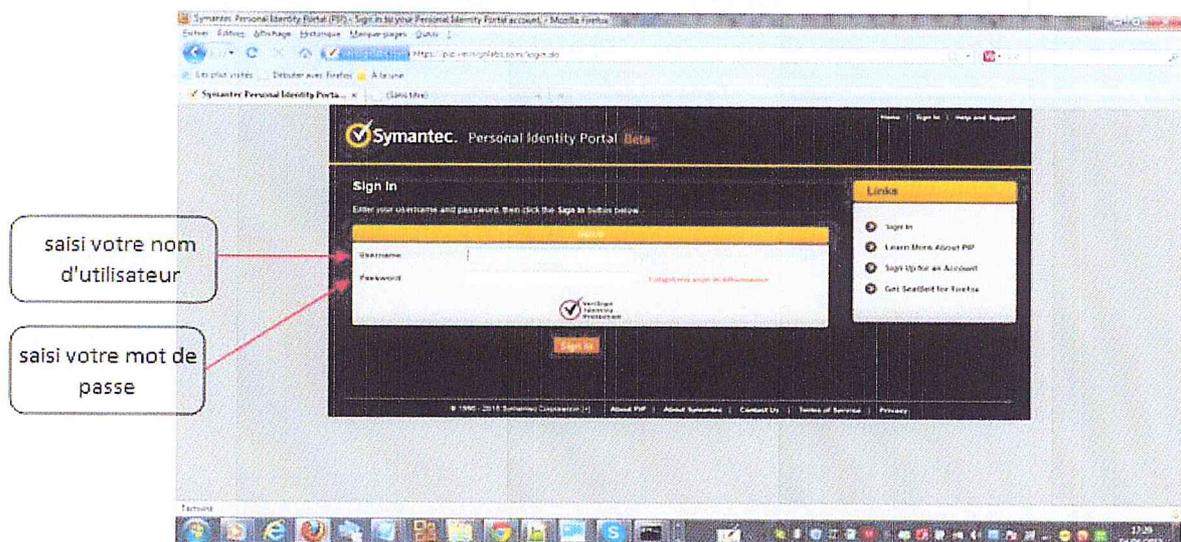


Figure 5.14: Vérification de l'authentification

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

Le résultat de l'authentification est:

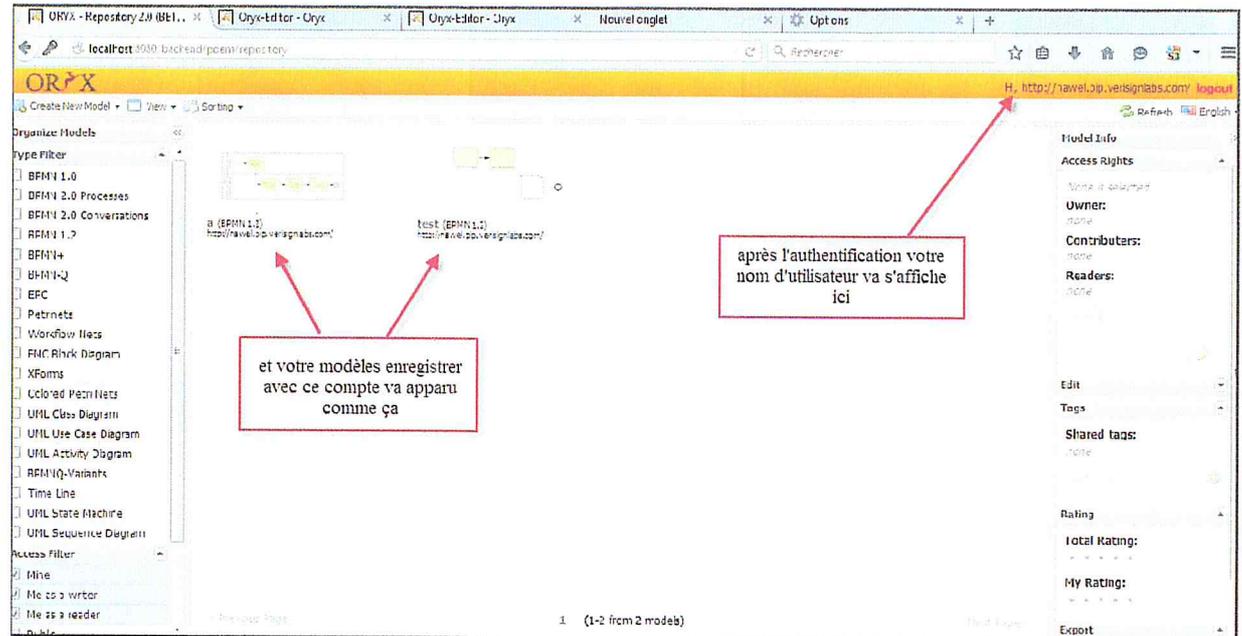


Figure 5.15: Résultat de l'authentification

## 2. Création d'un workflow inter-organisationnel

La figure ci-dessous représente un exemple sur un workflow inter-organisationnel.

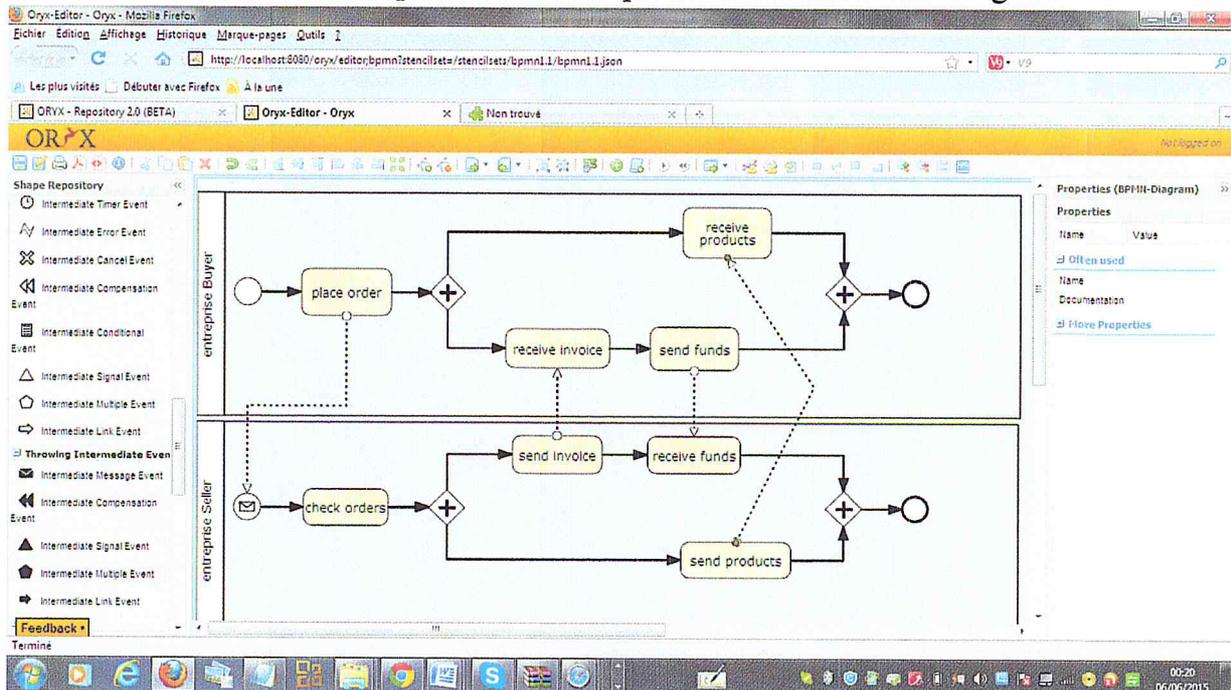


Figure 5.16: Workflow inter-organisationnel

## 3. Création d'annotations sémantiques :

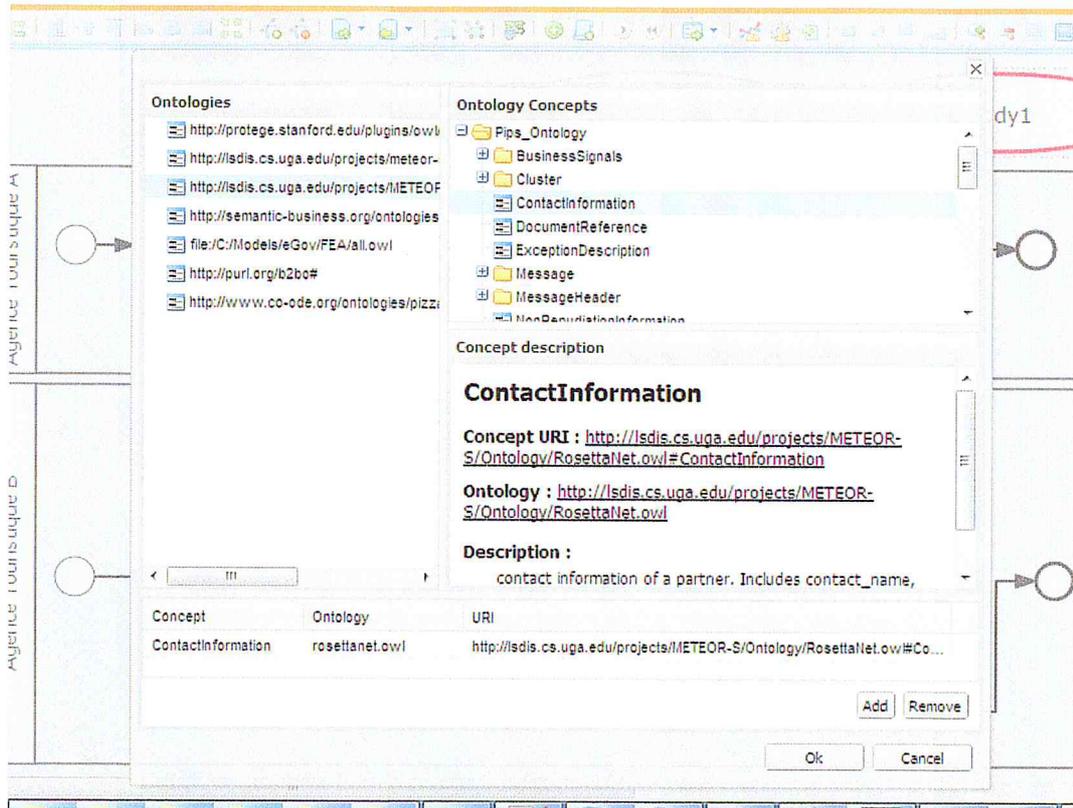
i. C'est un exemple d'une annotation d'une activité du Workflow:

pour illustrer le mécanisme d'annotation on va l'appliqué tout d'abord sur une Tache d'un Workflow; on commence par la schématisation de l'annotation par la suite l'attribution des propriétés nécessaires aux éléments (Annotation, Body et Target).

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

La propriété la plus importante à associer est celle de Body "HasURI" (où on affecte le concept tiré d'une ontologie), pour cela on utilise l'explorateur d'ontologie suivant:



Le Navigateur d'ontologies est divisé en 3 zones :

- 1) **Ontologies** : liste les différentes Ontologies disponibles dans le dépôt des ontologies.
- 2) **Ontologies Concepts** : affiche l'arbre des concepts d'une ontologie sélectionnée, elle permet de parcourir l'arbre et de choisir un concept.
- 3) **Concept description** : affiche la description de concept actuellement sélectionnée si elle est disponible.
- 4) **Tableau des concepts** : contient les annotations sémantiques qui sont déjà attribuées au symbole.
- 5) **Barre d'outils** : permet d'ajouter un concept au tableau des concepts, ou de supprimer un.

Les annotations sémantiques apparaissent sur le schéma comme montre la figure ci-dessous :

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

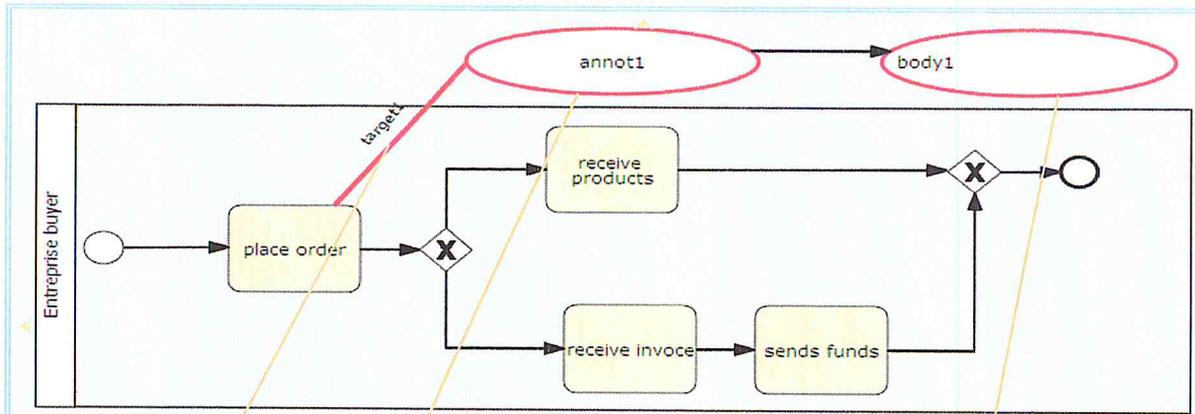
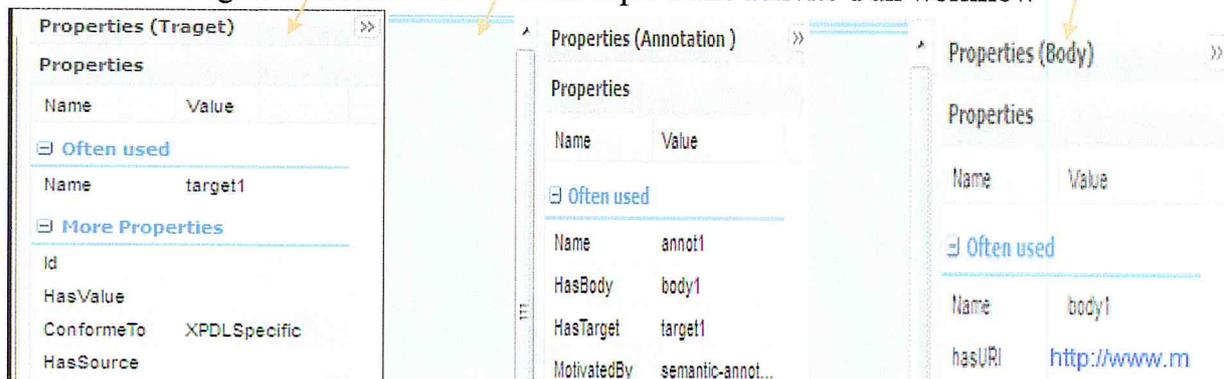


Figure 5.17: Annotation sémantique d'une activité d'un workflow



#### ii. Annotation de deux Activités des entreprises différente avec le même Body:

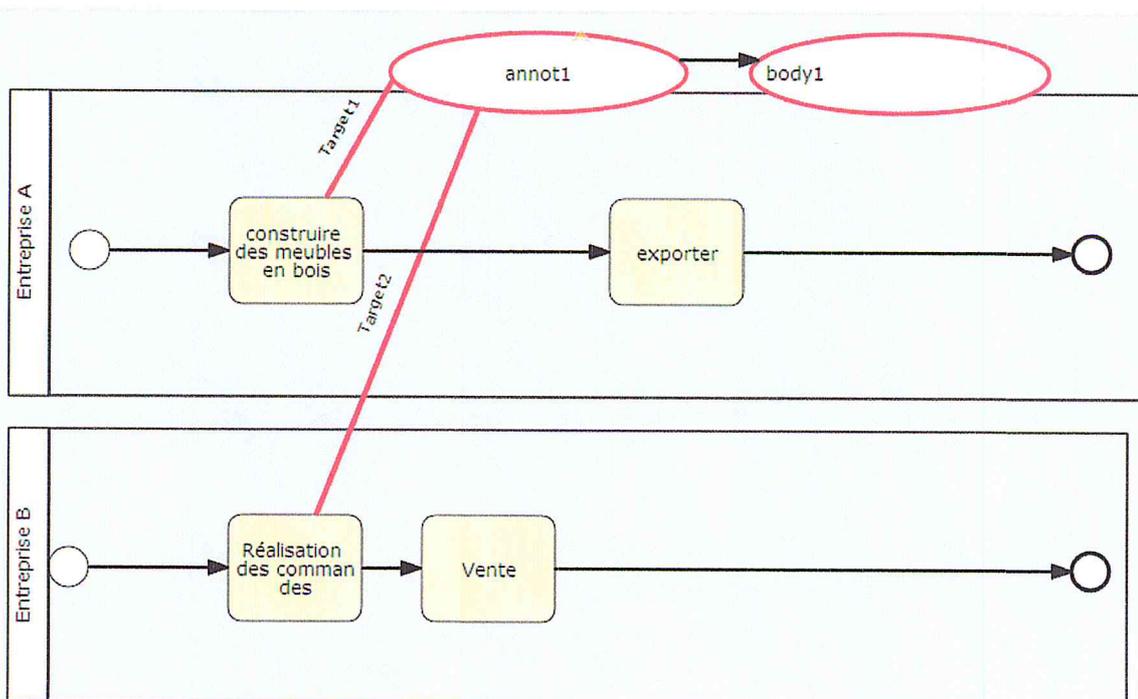


Figure 5.17: Annotation sémantique de deux activités d'un workflow inter organisationne

## Partie 02: Réalisation

### Chapitre 05: Description et mise en œuvre de la solution proposée

#### 4. Exporter via le nouveau plugin 'Export Open Annotation'

Ce fichier représente l'exemple d'annotation sémantique du l' activité annoté dans l'exemple 2.i ci-dessus.

```
<?xml version="1.0" encoding="UTF-8"?>
<Package Language="English" Id="oryx-canvas123">
  <Prefix base="http://www.w3.org/1999/02/22-rdf-syntax-ns#" oa4xpdL="http://www.cdta.dz/ns/oa4xpdL/" oa="http://www.W3C/ns/oa/" />
  <OpenAnnotations>
    <Annotation Id="oa4xpdL:oryx_67148B48-5974-4A8B-A16B-75FD986C98F0" Name="annot1" Type="oa:Annotation" hasBody="oa:body1"
      hasTarget="oa:target1" MotivatedBy="oa4xpdL:semantic-annotation" />
    <Body Id="oa4xpdL:oryx_EA178B76-2580-4828-AD76-3616E1EB06B4" Name="body1"
      HasUri="http://www.masfak.ni.ac.rs/ontologies/2009/8/SCOR-FULL.owl#quality"/>
    <SpTarget Id="oa4xpdL:oryx_D7DFCE8A-ED6C-45E7-ACC8-D8E2C85FFEDE" Name="target1" HasXpdLSelector="oa4xpdL:xpdLselector1"
      Type="oa4xpdL:XPDLSelector"
      HasValue="oa4xpdL:oryx_6A7FE6D9-1530-4609-A516-12483CB05395" ConformsTo="oa4xpdL:XPDLSpecific"
      HasSource="oa4xpdL:oryx_5BF2BC4B-08D2-4022-8C51-A69F405E7978-process" />
  </OpenAnnotations>
</Package>
```

Figure 5.18: Le fichier résultat via l'exportation

## I. Conclusion

A la fin de ce chapitre, on est arrivé à notre objectif principal qui sert à étendre notre éditeur utilisé avec l'outil d'annotation sémantique des workflows collaboratifs grâce à une nouvelle technologie "Open Annotation".

# *Conclusion Générale*

## Conclusion Générale

La problématique principale de notre travail a été de lever certaines contraintes posées lors de la collaboration des entreprises. Il se trouve que ces entreprises partenaires se voient obligées de relier leurs processus métier (workflows) pour implémenter cette collaboration.

Dans ce projet, nous avons eu l'occasion d'effectuer une étude d'état de l'art, sur la gestion des processus métier, les technologies workflow, ainsi que sur les technologies sémantiques, pour aboutir à la sélection des outils et méthodes nécessaires pour la réalisation de notre objectif d'interopérabilité des workflows inter-organisationnels.

Pour se faire, nous avons personnalisé l'éditeur open-source de workflows « Oryx editor », pour y intégrer un nouveau procédé d'annotation sémantique, qui permet d'attacher des descriptions sémantiques à des workflows syntaxiques.

Nous avons donc eu l'occasion de découvrir de nouvelles notions et technologies comme les langages de workflows BPMN, XPD L etc., les technologies sémantique et les ontologies, ainsi que différents mécanismes d'annotation sémantique comme SA-WSDL et surtout le modèle "Open Annotation". Il nous a aussi été donné de découvrir divers projets de la communauté open-source, comme "Oryx editor", ainsi que les difficultés qui y sont associées.

Ce travail a été une expérience très enrichissante, autant sur le plan pédagogique que personnel. Nos livrables vont être intégrés par le CDTA au sein d'une plateforme en cours de construction, qui vise à offrir des fonctionnalités supportant la collaboration interentreprises, selon une architecture Cloud.

Au sein de cette plateforme, notre travail permettra à des entreprises de modéliser leurs workflows et de travailler sur leur alignement au sein d'un workflow inter-organisationnel. Le procédé d'annotation, que nous avons mis en place, permettra d'explicitier les concepts manipulés par chaque partenaire, pour s'il le faut réaliser les traductions et les transformations nécessaires, pour garantir l'interopérabilité. La plateforme Cloud de collaboration interentreprises initiée par le projet du CDTA intégrera d'autres fonctionnalités relatives à la médiation sémantique, la découverte de partenaires, la vérification de la compatibilité comportementale, etc.

En Algérie, les entreprises et les administrations doivent nécessairement aller nécessairement vers l'ouverture de leurs systèmes d'information, pour collaborer avec des partenaires (Secteur de la Santé et celui de Sécurité sociale, les douanes et les impôts, la gendarmerie nationale et la sûreté nationale, ainsi que les entreprises économique et les PME/PMI, etc.). La mise en place de référentiels sectoriels (ontologies) et l'utilisation de technique de désambiguïsons par annotation, peuvent constituer un début de solution.

## Perspectives

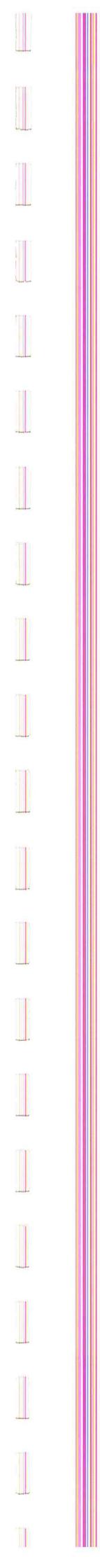
Les travaux présentés dans ce mémoire laissent envisager quelques perspectives qu'on présentera dans ce qui suit :

- l'intégration d'un nouvel outil de visualisations des ontologies "VOWL" pour faciliter la visualisation des concepts d'une ontologie.
- Faire enrichir la nouvelle ontologie crée OA4XDL avec de nouvelles classes dans le but d'améliorer l'annotation sémantique avec oa pour XPDL.
- établissement des règles de comparaison haut niveau entre les différents ontologies come "FOAF" ou bien "SAME AS" ... etc.
- continuation de la partie "mapping " qui contient l'étape de comparaison entre les différentes données des workflows inter-organisationnels et faire la conversion si c'est nécessaire.

## Liste des abréviations, sigles et acronymes

- BPMN:** Business Process Management Notation: un formalisme pour la représentation graphique des processus d'affaires .
- JSON :** JavaScript Object Notation.
- Plugiciel:** Logiciel d'application complémentaire qui associé à un fureteur internet, ajoute de nouvelles fonctionnalités. Plugiciel le terme recommandé par l'office québécois de la langue française pour le terme anglais : Plugin.
- SVG:** Scalable Vector Graphics.
- JS:** JavaScript.
- URI:** Uniform Resource Identifier.
- OA:** Open Annotation
- XPDL:** *XML Process Definition Language*
- OA4XPDL:** Open Annotation For XPDL.
- BPEL:** *Business Process Execution Language.*
- BPEL4WS:** *Business Process Execution Language for Web Service.*
- B2B:** Business to Business.
- WfMC:** Workflow Management Coalition.
- SGWf:** Systeme de Gestion de Workflow.
- WFIO:** Workflow inter-Organisationel.
- EPC:** Event Drive Process.
- OMG:** Object Management Group.
- UML:** *Unified Modeling Language.*
- WSFL:** Web Services Flow Language.
- XML:** Extensible Markup Language.
- IDE:** *Integrated Development Environmen.*
- BPMS:** Business Process Management System.
- SaaS:** *Software as a Service.*
- SOAP:** *Simple Object Access Protocol.*

- WSDL:** *Web Services Description Language.*
- UDDI:** *Universal Description Discovery and Integration.*
- HTTP:** HyperText Transfer Protocol.
- W3C:** World Wide Web Consortium.
- JSON-LD:** JavaScript Object Notation for Linked Data.
- OWL:** Web Ontology Language.
- SHOE:** Simple HTML Ontology Extension.
- HTM:** HyperText Markup Language.
- RDF:** *Resource Description Framework.*



# *Bibliographie*

## Bibliographie

(signavio-core-components) Consulté le avril 22, 2015, sur signavio-core-components:  
<https://code.google.com/p/signavio-core-components/>

(book) Consulté le mars 21, 2015, sur bpm-book: <http://www.bpm-book.com/>

Amardeilh, F. (2007). *Web Sémantique et Informatique Linguistique : propositions méthodologiques et réalisation d'une plateforme logicielle*. these de doctorat, Université Paris X –Nanterre.

Andre, T., & Tchamga, N. (2010). *La communication au service du management. cas de l'ETAP*. Master professionnel en management et stratégie des entreprises, Faculté méditerranéenne libre de Tunis, Tunis.

Baouab, A. (2013a). *Gouvernance et supervision decentralisee des choregraphies inter-organisationnelles*. these doctorat, Université de Lorraine, Lorraine.

Belaid, N. (2011a). *Modelisation de services et de workows semantiques a base d'ontologies de services et d'indexations. Application a la modelisation geologique*. these doctorat, Ecole Doctorale : Sciences et Ingénierie pour l'Information.

Belghait, F. (2011). *EXPLORATION DE LA MIGRATION DE LA BASE DE DONNÉE RELATIONNELLE DU SYSTÈME DE GESTION DE PROCESSUS D'AFFAIRE ORYX VERS LA BASE DE DONNÉE NO-SQL UTILISÉE PAR LA PLATEFORME DE L'INFORMATIQUE DE NUAGE DE HADOOP*. mémoire, UNIVERSITÉ DU QUÉBEC.

Benfifi, K. (2012a). *Une architecture à base d'agents pour le workflow inter-organisationnel lâche*. Mémoire de magister, Université UHL Batna.

Benhbireche, H. (2014). *Conception et réalisation d'un système intelligent de recherche d'information réglementaire*. Master en Informatique, UNIVERSITE KASDI MERBAH OUARGLA.

BITI, I., & ELBEZZAOUI, K. *WORKFLOW*. Nancy Université.

BOUHAJJA, L. (2012). *Implémentation d'une application Météo avec android*. universite de TUNIS EL MANAR.

BPMN, S. (2003). *BPMN and Business Process Management*.

*cat-logistique.com*. (s.d.). Consulté le avril 15, 2015, sur <http://www.cat-logistique.com/audit.htm>

Chabeb, Y. (2011). *Contributions à la Description et la Découverte de Services Web Sémantiques*. these doctorat, Université d'Évry Val d'Essonne.

Chebbi, I. (2007). *CoopFlow : une approche pour la coopération ascendante de workflows dans le cadre des entreprises virtuelles*. these doctorat, Université d'Évry Val d'Essonne.

Consortium, e. W. (2008). *Semantic hubs for geological projects* .

Corcho, O. (2006). *Ontology based document annotation: trends and open research problems*. *International Journal of Metadata, Semantics and Ontologies, Inders cience*.

(Open Annotation) Consulté le mai 2, 2015, sur site Web openannotation:  
<http://www.openannotation.org>

Dafinis, B. (2008). *XML process modeling for disruptive change planning: a case study of a newspaper circulation processes.*

Dardek, A., & Zouag, R. (2011). *Interopérabilité des services web« Application sur la réservation de vol»*. mémoire de Licence, Université Abou Bakr Belkaid, Tlemcen.

Death, S. (2011a). *service Web.*

*developpez.com.* (s.d.). Consulté le mars 22, 2015, sur <http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-activites>

Devillers, M. (2011). *The Business-IT.* thèse de master, Radboud University Nijmegen.

Donsez, S. M. (2010). *Advanced Sensors and lightweight Programmablemiddleware for Innovative Rfid Enterprise applications.*

ElMansouri, R. (2011). *Modélisation et Vérification des processus métiers dans les entreprises virtuelles : Une approche basée sur les entreprises virtuelles : Une approche basée sur les entreprises virtuelles : Une approche basée sur la transformation de graphe.* these doctorat, Université Mentouri Constantine.

Farquhar, A. R. (1997). *The ontolingua server: a tool for collaborative ontology construction.*

Fdhila, W. (2011a). *Decentralisation Optimisee et Synchronisation des Procedes Metiers Inter-Organisationnels.* these doctorat, universite Henri Poincare.

FODIL, B. (2011). *EXPLORATION DE LA MIGRATION DE LA BASE DE DONNÉE RELATIONNELLE .* UNIVERSITÉ DU QUÉBEC.

Franck, W. *Conception et développement d'une plate-forme de coopération inter-organisationnelle :le point de synchronisation.* Memoire INGENIEUR C.N.A.M., CENTRE REGIONAL DE NANCY.

Gaaloul, K. (s.d.). Développement et intégration des techniques de Data Mining pour une plate-forme de coopération. Tunis: Ecole national d'ingéierie de Tunis.

Gaaloul., W. (2006). *La Découverte de Workflow Transactionnel pour la Fiabilisation.* these doctorat, Université Henri Poincaré.

Gruber. (1993).

Helga, D. (2007). *Canevas pour la composition de services web avec.* these doctorat, Université Joseph Fourier.

Hernandez, N. (2005). *Ontologies de domaine pour la modélisation du contexte en Recherche d'information.* these doctorat, Université Paul Sabaltier de Toulouse.

Jonuzi, V. (2014a). *Portability benchmark of XPDL file among WfMSs from different vendors.* these master , École de technologie de l'information Politecnico di Milano.

Katiri, M. Y. (2010). *LA MODÉLISATION DES PROCESSUS D'AFFAIRES EN UTILISANT BPMN AINSI QUE QUALIGRAMME : ÉTUDE DE CAS D'UN PROJET D'IMPLÉMENTATION D'UN ERP*. UNIVERSITÉ DU QUÉBEC.

LEFEBVRE, B. (2003). *DÉVELOPPEMENT UN OUTIL EFFICACE POUR ANNOTER DES DOCUMENTS*. Mémoire de fin d'études, Institut de la Francophonie pour l'Informatique.

Marek, L. (2006). *Data Access in Workflow Management Systems*. Berlin.

Mille, D. (2005). *Modèles et outils logiciels pour l'annotation sémantique de documents*. these de doctorat, Université Joseph-, Département informatique.

Mohan, C. (1999). *Workflow management in the Internet Age*. école d'été d'objet repartis.

Nomane, O. A. (2008). *Une approche sémantique pour la description, l'abstraction et l'interconnexion de workflows dans un contexte inter-organisationnel*. these de doctorat, Université d'Évry Val d'Essonne.

Peters, N. (2009). *Oryx -Introduction*.

Rademakers, T. *Activity in action*.

Sini, G. (2013). *les aspects opérationnels d'un processus : la séquence des tâches et qui les réalisent, le flot de données qui supporte ces tâches, et les mécanismes qui permettent de mesurer, suivre et contrôler ces tâches*. these doctorat, universite MOULOUD MAMMERI, TIZI OUZOU.

Soussi, A. (2013a). *Modélisation centrée sur les processus métier pour la génération complète de portails collaboratifs*. these doctorat, Universit Lille1.

Specif. (s.d.). Consulté le mai 2, 2015, sur <http://www.openannotation.org/spec/core/specific.html#Specific>

Stephen A. White, S. (s.d.). *XPDL AND BPMN*. United States.

Touzi, J. (2007). *Aide à la conception de Système d'Information Collaboratif support de l'interopérabilité des entreprises*. these de doctorat, INSTITUT NATIONAL POLYTECHNIQUE DE TOULOUSE.

White, S. (2004a). *Introduction to BPMN*.

Xavier, G. (2009). *Urbanisation et architecture des SI*. cours.

Zefouni, S. (2012). *Aide à la conception de workflows personnalisés : application à la prise en charge à domicile*. these doctorat, UNIVERSITÉ DE TOULOUSE, TOULOUSE.

Zouhel, B. (2008a). *APPARIEMENT SÉMANTIQUE DES CVs/OFFRES D'EMPLOI DANS LE CADRE DU E-RECRUTEMENT*. Mémoire, Université Mentouri de Constantine.

# *Annexe*

**ANNEXE A: CONFIGURATION DE  
L'ENVIRONNEMENT DE  
DÉVELOPPEMENT**

## I. Création d'environnement de travail d'oryx

Ce annexe montre comment installer et configurer l'environnement de développement pour construire l'application Oryx. Pour cela il faut tout d'abord installer certains outils dans un ordre précis suivi par un ensemble de configuration :

### A. Les outils nécessaires:

- 1) Java JDK 5 ou plus du lien <http://java.sun.com/javase/downloads/index.jsp> suivi des configurations suivantes:
  - Assurez-vous que la variable d'environnement "JAVA\_HOME" est réglé sur java l'installation dir (par exemple "C: Program files \ de \ java \ jdk1.6").
  - Veiller à ce que la variable d'environnement PATH est étendue avec "\ bin" - folder de java l'installation dir (par exemple "C: Program files \ de \ java \ jdk1.6 \ bin")
- 2) Le navigateur Firefox via le lien <http://www.mozilla.com/de/firefox/>.
- 3) Le serveur Tomcat à partir du lien <http://tomcat.apache.org> . Nous avons utilisé la version 6 du serveur Tomcat.
- 4) Eclipse IDE for Java EE Developers a partir de (<http://www.eclipse.org/downloads/> ).
  - Vous devez également installer un plugin pour eclipse qui est subversion, par le lien <http://subclipse.tigris.org/>.
  - Intégrer un Aptana-plugin.
  - Dans eclipse, checkout Oryx a partir de SVN tel que :
  - Sélectionnez "Fichier -> Importer ..." et choisissez "SVN -> Projet de SVN"
  - Suivez l'assistant. Créer un nouvel emplacement dépôt en utilisant le lien disponible ici <http://code.google.com/p/oryx-editor/source/checkout> Puis checkout le "trunk\_folder".
- 5) Installer la version Python 2.5.2 via <http://www.python.org/download/releases/2.5.2>.
- 6) PostgreSQL 8.3.x (IMPORTANT: utiliser exactement cette version)
  - Dans l'assistant d'installation sélectionnez "Installer PostgreSQL comme un service".
  - Veiller à ce que PATH variable d'environnement est étendue avec "\ bin" - folder de PostgreSQL l'installation dir (par exemple "C: \ program files \ PostgreSQL \ 8.3 \ bin")
  - changer de répertoire de fichiers de schéma qui se trouve dans " oryx-workspace/poem-jvm/data/database/db\_schema.sq "

- Exécutez les commandes suivantes pour importer schéma:
  - createuser -U postgres -e -P -E -S -d -r poem
    - le mieux est d'utiliser un mot de passe poem
    - entrez le mot de passe pour «poem» deux fois (celui qui est spécifié dans oryx/poem-jvm/etc/hibernate.cfg.xml).
    - entrez le mot de passe utilisateur postgres (Windows uniquement )
  - createdb -U postgres -e -E utf8 -O poem poem
    - entrez le mot de passe utilisateur postgres.
  - psql -U postgres poem <C:/Users/...../workspace\_folder/oryx/poem-jvm/data/database/db\_schema.sql>.
    - entrez le mot de passe utilisateur postgres.

## B. Les configurations nécessaire:

- Ouvrez le fichier build.properties dans répertoire racine, et edit "deploymentdir" et réglez à votre Apache Tomcat webapps dossier (par exemple C: /apache-tomcat-6.0.32/webapps).
- Construire Oryx pour produire de fichiers "war-file" déployables:
  - Clic droit sur "build.xml" dans répertoire racine et sélectionnez " Run As -> External Tool Configuration"
  - Réglez buildfile à "build.xml", par exemple \$ {workspace\_loc: /oryx/build.xml}
  - Réglé répertoire de base sur le répertoire racine oryx, par exemple \$ {workspace\_loc: / oryx}
  - Choisissez l'onglet " targets " et vérifier les objectifs suivants: «build-all» et «deploy all»
  - Lancer la construction en cliquant sur le bouton-exécuter.
  - Si la construction était faite avec succès, les deux fichiers "oryx.war" et "backend.war" auraient été copiés sur votre Apache Tomcat dossier "\ webapps".

## C. Lancement de Oryx

- Démarrer Apache Tomcat avec "apache-install-dir / bin / startup.bat".
- Pour le test lancez votre navigateur et "ouverte <http://localhost:8080/backend/poème/repository> ".
- Maintenant, tout fait.

## Annexe B : BPMN1.2

Dans cette partie, nous allons présenter les principaux éléments du langage de modélisation BPMN (Business Process Management Notation) dans sa version 1.2.

## I. Les objets graphiques de BPMN

La notation BPMN définit un ensemble d'objets graphiques qui permettent la description des processus. En effet, cette notation se base sur 4 familles d'objet :

- Objets de flux.
- Objets de connexion.
- Couloirs d'activités.
- Artefacts.

Chaque famille d'objets contient des sous catégories d'objets. Les sections suivantes vous introduisent aux concepts d'objets utilisés dans la notation BPMN.

### I.1. Objet de flux

Sont les principaux éléments graphiques pour définir le comportement des processus d'une entreprise:

1. Événements: ces événements affectent le flux du processus et ont habituellement une cause (déclencheur) ou un impact (résultat). Les événements sont des cercles avec les centres ouverts pour permettre de différencier les marqueurs internes des différents déclencheurs ou des résultats. D'une manière générale, on distingue 3 types d'événements tels qu'illustrés dans le tableau ci-dessous :

Type d'événement	Symbole	Description
Début		Événement de départ d'un processus.
Intermédiaire		Événement qui se déroule lors de l'exécution d'un processus.
Fin		Événement de fin d'un processus.

Tableau1. Type d'événements BPMN (BPMN, 2003)

2. Activités: Une activité est un terme générique pour le travail que l'entreprise effectue. Une activité peut être atomique ou non atomique (composée). La notation BPMN distingue entre deux types d'activités:
  - ❖ Tâche : Une action indivisible
  - ❖ Sous-processus : Une action qui regroupe un certain nombre de détails.
  - ❖

Le tableau suivant présente les symboles utilisés pour illustrer ces deux types d'activités :

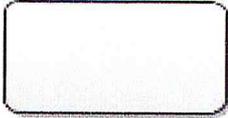
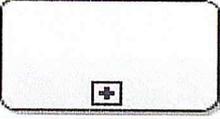
Type d'activité	Symbole
Tâche	
Sous-Processus	

Tableau 2. Types d'activités BPMN (Katiri, 2010)

3. Le pont: est utilisé pour contrôler la divergence et la convergence des multiples flux de séquence. Les passerelles sont représentées par un losange vide dans le cas d'une décision classique. Sinon, un signe supplémentaire pour indiquer des cas plus complexes.

### I.2. Objet de connexion

Ont pour objectif de fournir des moyens de relier les objets de flux à l'un à l'autre. Les trois types d'objets de connexion utilisés par la notation BPMN sont:

1. Flux séquentiel: est utilisée pour montrer l'ordre selon lequel les activités seront réalisées dans un Processus.
2. Flux de message: est utilisé pour montrer le flux de messages entre deux participants.
3. Flux d'association: Une association est utilisée pour associer des informations avec un objet de flux. Texte et graphique non- objet de flux peuvent être associés avec les objets de flux.

Le tableau ci-dessous présente les trois types d'objets de connexion utilisés par la notation BPMN.

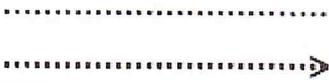
Objet de connexion	Symbole
Flux séquentiel	
Flux d'association	
Flux de message	

Tableau 3. Objets de connexion BPMN (White, 2004a)

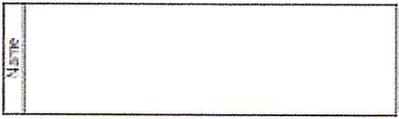
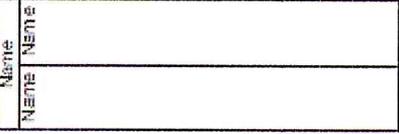
### I.3. Les couloirs d'activités

Sont utilisés pour regrouper les éléments de modélisation primaires: Bassin ou piste (Pool) et Couloir ou corridor (Lane).

1. Pool: Il sert à représenter un processus bien déterminé, en structurant les tâches selon le participant qui les exécute. Ce composant peut contenir plusieurs couloirs.

2. **Lane:** Conteneur graphique qui contient les activités d'un participant (rôle) dans un processus.

Ce tableau fournit des explications sur les deux types de couloirs proposés par cette notation.

Type de couloir d'activités	Représentation graphique
Lane	
Pool	

Tab 4. Les types de couloirs d'activités en BPMN (White, 2004b)

#### I.4. Artefacts

Sont des objets supplémentaires qui apportent des détails complémentaires afin d'assurer une compréhension parfaite du processus, les trois types d'artefacts de la notation BPMN sont :

1. **Annotation:** C'est un objet qui contient des commentaires, que le modelleur peut ajouter dans le diagramme BPMN.
2. **Groupe:** C'est un objet utilisé pour regrouper plusieurs tâches.
3. **Objet de données:** Objets utilisés pour représenter les données nécessaires pour l'exécution d'une activité, ainsi que celles produites. La liaison de cet objet avec la tâche concernée se fait par un flux d'association.

Le tableau suivant présente les trois types d'artefacts de la notation BPMN, ainsi les symboles associés.

Artefact	Symbole
Annotation	
Groupe	
Objet de données	

Tab 5. Les artefacts BPMN (White, 2004c)

## Annexe C: XPDL 2.1

XPDL (XML Process Definition Language) est un langage de description de workflow standardisé par le WfMC (Workflow Management Coalition). Il est utilisé dans un grand nombre de système de gestion de workflows. Plus généralement, la WfMC promeut et développe l'utilisation de la technologie de Workflow, grâce à l'établissement de normes pour constituer une terminologie commune, améliorer l'interopérabilité et la connectivité entre les systèmes de gestion de workflow.

## I. XPDL 2.1

En Avril 2008, la WfMC a ratifié XPDL 2.1 comme la quatrième révision de cette spécification. XPDL 2.1 inclut des extensions à gérer les nouvelles constructions BPMN 1.1 ainsi que la clarification des critères de conformité pour les implémentations, et est destiné à être utilisé comme un format de sérialisation pour BPMN.

XPDL fournit un format de fichier XML qui peut être utilisé pour échanger des modèles de processus entre les outils de modélisation et même d'exécution.

## II. Méta- Modèle de XPDL pour la définition des processus

D'une manière générale, un méta modèle décrit la grammaire du langage de modélisation ainsi son utilisation. (Dafinis, 2008)

Le méta-modèle XPDL décrit les entités de haut niveau contenues dans une définition de processus, leurs relations et les attributs. Il est représenté dans la **Error! Reference source not found**. Tel qu'il est proposé par WfMC.

Dans ce méta-modèle un Process (*Workflow*) possède plusieurs attributs (*Data Field*) et contient plusieurs activités, transitions et participants (*humains, applications*).

Une activité est liée à une tâche dans le workflow. Les transitions assurent la liaison entre les activités qui peuvent accéder aux Data field (données) du workflow.

## III. Les composants de XPDL

XPDL permet la définition d'un workflow basé sur une syntaxe XML comportant les principaux éléments suivants :

- les processus servent à définir des workflows ou des parties de workflow.
- les activités qui sont des concepts de base dans XPDL et il en existe trois types:
  1. Les activités de routage (Route) qui sont utilisées pour exprimer les contrôles de flux complexes entre des activités qui ne peuvent être exprimés seulement avec des transitions.
  2. Les activités de bloc (BlockActivity) qui servent à exécuter un ensemble d'activités.
  3. Les activités de type implémentation (Implementation) qui servent à exprimer des tâches du workflow.
- Les transitions sont les relations entre les activités ;

- les participants décrivent les entités qui vont réaliser les tâches des activités. Un participant peut en effet être soit une ressource ou un ensemble de ressources, une organisation, un rôle, un être humain ou un système.

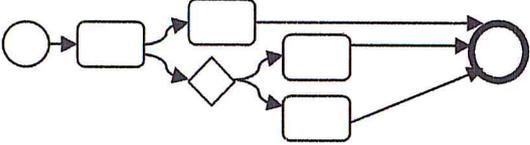
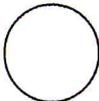
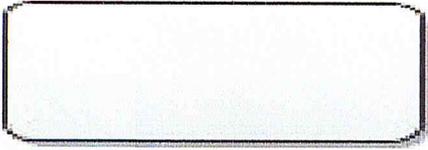
#### IV. XPDL et BPMN

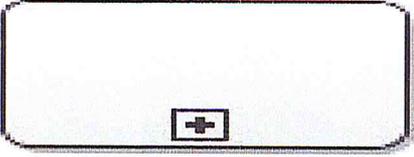
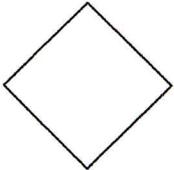
BPMN (Business Process Modeling Notation) est un standard de notation visuelle de processus approuvée par le WfMC, et largement adopté dans toute l'industrie. Jusqu'à la version 1.2 de la norme BPMN il n'y est défini que l'apparence de sous laquelle le processus s'affiche à l'écran. La façon que vous stockez et échangez ces définitions de processus était en dehors du champ d'application, et ce est là XPDL entre en jeu.

XPDL fournit un format de fichier qui prend en charge tous les aspects de la notation de définition de processus BPMN y compris les descriptions graphiques du diagramme, ainsi que les propriétés utilisées au moment de l'exécution. *Et donc XPDL est le format de sérialisation pour BPMN. (Donsez, 2010)*

#### V. Correspondance de BPMN à XPDL

Du point de vue structurel, BPMN et XPDL sont très similaires. En fait, la mise en correspondance entre BPMN et XPDL est beaucoup plus simple que la correspondance entre BPMN et BPEL4WS ou BPML. Nous voyons quelque mapping de BPMN à XPDL dans le tableau suivant:

<i>Objets graphiques de BPMN</i>	<i>Mappage à XPDL</i>
	<WorkflowProcess/>
Start event 	<Activity> <Route/> </Activity>
	<Transition/>
Task 	<Activity> <Implementation> <Tool/> <Performer/> </Implementation> </Activities>

 <p>Sub-Process</p>	<pre>&lt;Activity&gt; &lt;Implementation&gt; &lt;SubFlow/&gt; &lt;/Implementation&gt; &lt;/Activities&gt;</pre>
 <p>Decision</p>	<pre>&lt;Activity&gt; &lt;Route/&gt; &lt;TransitionRestriction&gt; &lt;Split Type="XOR"/&gt; &lt;/TransitionRestriction&gt; &lt;/Activities&gt; Combined with a: &lt;Transition&gt; &lt;Condition/&gt; &lt;Transition&gt;</pre>
 <p>End Event</p>	<pre>&lt;Activity&gt; &lt;Route/&gt; &lt;/Activity&gt;</pre>

Tab1. Objets BPMN et leurs mappages à XPD (Stephen A. White, s.d.)

## VI. Exemple d'un workflow XPD

L'exemple suivant affiche un code XPD qui représente une partie d'un processus.

## Annexe E : Ontologie OA4XPDL

```
65
66 <owl:ObjectProperty rdf:about="{oa4xpd1;conformsto}"
67   <rdfs:label xml:lang="en">conformsto</rdfs:label>
68   <rdfs:comment xml:lang="en">**** </rdfs:comment>
69   <rdfs:isDefinedBy rdf:resource="http://www.cdta.dz/ns/oa4xpd1#"/>
70   <rdfs:range rdf:resource="{oa4xpd1:XPDLSelector"/>
71   <rdfs:domain rdf:resource="{oa4xpd1:SpecificResource"/>
72 </owl:ObjectProperty>
73
74 <!-- http://www.w3.org/ns/oa4xpd1#hasValue -->
75
76 <owl:ObjectProperty rdf:about="{oa4xpd1;hasValue}"
77   <rdfs:label xml:lang="en">hasValue</rdfs:label>
78   <rdfs:comment xml:lang="en">The Id of the annotated task </rdfs:comment>
79   <rdfs:isDefinedBy rdf:resource="http://www.cdta.dz/ns/oa4xpd1#"/>
80   <rdfs:range rdf:resource="{oa4xpd1:XPDLSelector"/>
81   <rdfs:domain rdf:resource="{oa4xpd1:SpecificResource"/>
82 </owl:ObjectProperty>
83
84 <!-- http://www.cdta.dz/ns/oa4xpd1#motivatedBy -->
85
86 <owl:ObjectProperty rdf:about="{oa4xpd1;motivatedBy}"
87   <rdfs:label xml:lang="en">motivatedBy</rdfs:label>
88   <rdfs:comment xml:lang="en">The relationship between an Annotation and a Motivation, indicating the reasons why the Annotation was created.
89
90 Each Annotation SHOULD have at least one oa4xpd1:motivatedBy relationship, and MAY be more than 1.</rdfs:comment>
91   <rdfs:isDefinedBy rdf:resource="http://www.cdta.dz/ns/oa4xpd1#"/>
92   <rdfs:domain rdf:resource="{oa4xpd1:Annotation"/>
93   <rdfs:range rdf:resource="{oa4xpd1:SemanticAnnotation"/>
94 </owl:ObjectProperty>
95
96 <!--
97
98 //
99 // Classes
100 //
101 ////////////////////////////////////////////////////////////////////
102 -->
103 <!-- http://www.cdta.dz/ns/oa4xpd1#SemanticAnnotation -->
104
105 <owl:Class rdf:about="{oa4xpd1;SemanticAnnotation}"
106   <rdfs:label xml:lang="en">SemanticAnnotation</rdfs:label>
107   <rdfs:subClassOf rdf:resource="{oa4:motivation"/>
108   <rdfs:comment xml:lang="en">The SemanticAnnotation for creating an Annotation, indicated with oa4xpd1:motivatedBy, is a reason for its creation, and mi
109
110 Each Annotation SHOULD have at least one oa4xpd1:motivatedBy relationship to an instance of oa4xpd1:Motivation, which is a subClass of skos:Concept.</rdfs:comm
111   <rdfs:isDefinedBy rdf:resource="http://www.cdta.dz/ns/oa4xpd1#"/>
112 </owl:Class>
113
114 <!-- http://www.cdta.dz/ns/oa4xpd1#XPDLSelector -->
115
116 <owl:Class rdf:about="{oa4xpd1:XPDLSelector}"
117   <rdfs:label xml:lang="en">XPDLSelector</rdfs:label>
118   <rdfs:subClassOf rdf:resource="{oa4:Selector"/>
119   <rdfs:comment xml:lang="en">A resource which describes the segment of interest in a representation of a Source resource, indicated with oa4xpd1:hasXPDL
120
121 This class is not used directly in Annotations, only its subclasses are.
122
123 The nature of the Selector will be dependent on the type of the representation for which the segment is conveyed. The specific type of selector should be indic
124
125 The Specifierisapos;s description MAY be conveyed as an external or embedded resource (cnt:Content), or as RDF properties within the graph. The description SHOU
126   <rdfs:isDefinedBy rdf:resource="http://www.cdta.dz/ns/oa4xpd1#"/>
127 </owl:Class>
128
129
130 <!-- http://www.w3.org/ns/oa4xpd1#conformsto -->
```

```

<xpd12:Package xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:simulation="http://www.tibco
" http://www.tibco.com/XPD/xpdExtension1.0.0" xmlns:xpd12="http://www.wfmc.org/2004/XPDL2.0alpha" xsi:s
http://www.wfmc.org/standards/docs/TC-1025_bpmnxpd1_24.xsd" Id="_hs1XkBzfEeCITYFVvDH41g" Name="Process
  <xpd12:PackageHeader>
  <xpd12:RedefinableHeader PublicationStatus="UNDER_REVISION">
  <xpd12:Pools>
  <xpd12:WorkflowProcesses>
    <xpd12:WorkflowProcess Id="_ce1ZoBzfEeCITYFVvDH41g" Name="Process">
      <xpd12:ProcessHeader/>
      <xpd12:RedefinableHeader/>
      <xpd12:Activities>
        <xpd12:Activity Id="_mUx04BzfEeCITYFVvDH41g" Name="Generate Email">
          <xpd12:Implementation>
            <xpd12:Task>
              <xpd12:TaskService xpdExt:ImplementationType="WebService" Implementation="WebService">
            </xpd12:Task>
          </xpd12:Implementation>
          <xpd12:NodeGraphicsInfos>
        </xpd12:Activity>
        <xpd12:Activity Id="_y2r1YBzfEeCITYFVvDH41g">
        <xpd12:Activity Id="_zz7QvBzfEeCITYFVvDH41g">
        <xpd12:Activity Id="_79xrfSbzfEeCITYFVvDH41g" Name="Start">
        <xpd12:Activity Id="_8kEQsBzfEeCITYFVvDH41g" Name="End">
        <xpd12:Activity Id="_GQi9EBzqEeCITYFVvDH41g" Name="Approve by Manager">
        <xpd12:Activity Id="_Gtw1gBzqEeCITYFVvDH41g" Name="Approve by Director">
          <xpd12:Implementation>
            <xpd12:Task>
          </xpd12:Implementation>
          <xpd12:NodeGraphicsInfos>
        </xpd12:Activity>
      </xpd12:Activities>
      <xpd12:Transitions>
    </xpd12:WorkflowProcess>
  </xpd12:WorkflowProcesses>
  <xpd12:ExtendedAttributes>
</xpd12:Package>

```

Figure: Exemple: code XPDL pour le début du processus de vote par Email (Stephen A. White, s.d.)