

Olivier Glassey  
Jean-Loup Chappelet

**Comparaison de trois techniques de  
modélisation de processus:  
ADONIS, OSSAD et UML**

Working paper de l'IDHEAP 14/2002  
UER: Management public / Systèmes d'information

# **Comparaison de trois techniques de modélisation de processus: ADONIS, OSSAD et UML**

Olivier Glassey  
Jean-Loup Chappelet

UER: Management public / Systèmes d'information

Working paper de l'IDHEAP no 14/2002  
novembre 2002

*Ce document se trouve sur notre site Internet: <http://www.idheap.ch/>*

© 2002 IDHEAP, Chavannes-près-Renens

## ***Résumé***

Dans ce travail nous proposons de comparer trois techniques de modélisation de processus afin de déterminer quels en sont les concepts communs et d'identifier les différences majeures, s'il y en a. Nous avons retenu les techniques suivantes : ADONIS, un logiciel de modélisation de processus opérationnel qui intègre sa propre technique de modélisation ; OSSAD, une méthode de modélisation de système d'information et d'organisation ouverte et standard ; UML, le langage de notation graphique standardisé bien connu. Après avoir étudié dans le détail les concepts et les champs d'application propres à chacune de ces techniques, nous avons établi un comparatif qui se résume en trois interrogations : quels sont les objectifs de l'organisation ? quelle est la structure de l'organisation et quelles sont les ressources humaines disponibles ? comment fonctionne l'organisation ? Si les réponses à cette dernière question (comment ?) sont très similaires dans les trois cas, il ressort nettement que les différences les plus importantes entre ces techniques se situent au niveau de la modélisation qui relève des deux premières questions (quoi et qui ?).

## ***Abstract***

In this work we want to compare three process modelling techniques in order to find their common concepts and to identify their important differences, if there are any. We chose the following techniques: ADONIS, a software tool for operational process modelling that has its proprietary modelling technique; OSSAD, an open and standard method for information system and organization modelling; UML, the well-known standardized graphical notation language. We studied in detail the concepts and the application fields of each of these techniques and we established a comparison that can be summarized in three questions: what are the objectives of the organization? what are the structure and the human resources of the organization? how does the organization work? Although the answers to the latter question (how?) are very similar in the three cases, it is very clear that the main differences between these techniques appear at the level of modelling concerned by the first two questions (what and who?).

## Table des matières

1. Introduction	1
1.1 ADONIS	1
1.2 OSSAD	2
1.3 UML	3
2. Types de modèles et notation graphique	4
2.1 ADONIS	4
2.1.1 Carte de processus	4
2.1.2 Modèle d'environnement de travail	5
2.1.3 Modèle de processus opérationnel	7
2.2 OSSAD	11
2.2.1 Modèle abstrait	11
2.2.2 Modèle de rôles	12
2.2.3 Modèle de procédures	14
2.2.4 Modèle d'opérations	16
2.2.5 Matrice activités-rôles	17
2.2.6 Modèle d'unités organisationnelles	18
2.3 UML	20
2.3.1 Diagramme de cas d'utilisation	20
2.3.2 Diagramme de séquence	22
2.3.3 Diagramme de collaboration	23
2.3.4 Diagramme d'activité	24
3. Comparaison ADONIS-OSSAD-UML	26
3.1 Fonctionnement de l'organisation	26
3.2 Structure et ressources de l'organisation	27
3.3 Objectifs de l'organisation	28
3.4 Concepts de modélisation	28
4. Conclusion	29
5. Bibliographie	30
Liste des Working paper de l'IDHEAP déjà parus	31

# 1. Introduction

Tant dans le monde de l'entreprise que dans le milieu académique il existe de nombreuses techniques de modélisation de processus, qu'il s'agisse de méthodes, de langages de modélisation ou d'outils informatiques intégrés. Certaines d'entre elles sont ouvertes et gratuites, d'autres sont propriétaires et payantes. Devant ce large choix, nous avons décidé de comparer trois de ces techniques, que nous estimons représentatives de différentes «écoles» de la modélisation de processus. Cependant nous n'effectuons pas ce travail pour établir quelle est la «meilleure» d'entre elles, nous cherchons plutôt à déterminer quels en sont les concepts communs et à découvrir s'il en existe qui sont fortement différenciés (et si oui, pour quelles raisons). Nous nous intéressons également aux différents types de modèles offerts par ces méthodes et aux niveaux d'abstraction proposés. Ce travail est réalisé dans le cadre du projet européen eGOV<sup>1</sup> qui vise, entre autres, à préciser une méthode de réorganisation de processus dans les administrations publiques.

Pour le présent comparatif, nous avons choisi trois techniques de modélisation, respectivement Adonis, OSSAD et UML. Comme nous l'avons mentionné plus haut, nous pensons qu'elles sont représentatives de différentes tendances actuelles du domaine :

- Adonis est un logiciel de modélisation de processus opérationnel qui intègre sa propre **technique de modélisation** et son **langage de description propriétaires**.
- OSSAD est une **méthode de modélisation de système d'information et d'organisation ouverte et standard**. Elle a été développée dans le cadre d'un projet de recherche européen et il existe quelques outils la supportant, dont OSS@D Process Design que nous avons utilisé pour ce comparatif.
- UML est un **langage de notation graphique standardisé** qui peut être utilisé dans différents domaines de modélisation et notamment pour décrire des processus de gestion. Il existe des dizaines de logiciels qui supportent UML, dont le plus connu est Rational Rose. Pour ce travail nous avons utilisé UMLStudio.

Aux paragraphes suivants nous présentons brièvement ces trois techniques, avant de les étudier individuellement et dans le détail au chapitre 2. Nous verrons en effet les différents types de modèles que chacune propose, en étudiant de près leurs notations graphiques respectives et en les illustrant par des exemples. Comme ces techniques ont des champs d'application relativement divers, nous nous concentrerons uniquement sur la modélisation de processus. En outre, nous ne parlerons pas des outils qui les supportent et des fonctionnalités que ces derniers offrent, notamment en terme d'analyse, de simulation, d'implémentation d'applications ou de génération de documentation. Au chapitre 3 nous passerons ensuite à la comparaison à proprement parler des concepts proposés par Adonis, OSSAD et UML.

## 1.1 ADONIS

Contrairement à OSSAD et UML<sup>2</sup> qui sont des techniques ouvertes et standardisées, Adonis est une méthode propriétaire totalement intégrée avec l'outil du même nom. Ce dernier a été développé par l'entreprise Business Objects Consulting (BOC)<sup>3</sup>, une société spécialisée dans

---

<sup>1</sup> Voir [www.egovproject.org](http://www.egovproject.org)

<sup>2</sup> Voir points 1.2 et 1.3

<sup>3</sup> Voir [www.boc-eu.com](http://www.boc-eu.com)

la gestion de processus opérationnels et issue du groupe BPMS (Business Process Management Systems) de l'Institut d'Informatique et d'Economie de l'Université de Vienne. Adonis est largement utilisé dans les services financiers et dans les administrations publiques, en particulier pour l'optimisation et la documentation de processus ainsi que pour la gestion de la qualité ou les certifications de type ISO. Il contient un module d'acquisition de données qui permet par exemple d'importer des statistiques en format Excel ainsi qu'un module de modélisation graphique. Celui-ci supporte uniquement la méthode standard Adonis dans sa version de base, mais il est possible d'acquérir des modules supplémentaires intégrant par exemple la méthode IBM Lovem ou certains diagrammes d'UML. Dans ce travail nous nous intéresserons uniquement à la technique de modélisation propre à Adonis. L'outil Adonis permet par ailleurs d'effectuer des analyses poussées grâce à son langage d'interrogation intégré et de procéder à des simulations basées sur des algorithmes de calcul de chemin, de temps ou de coûts.

La modélisation standard d'Adonis repose sur trois types de modèles :

- La *carte des processus* donne une idée générale des différents processus ou sous-processus effectués au sein d'une organisation.
- Le *modèle d'environnement de travail* permet de représenter la structure d'une organisation en termes d'unités, de responsables et de rôles. Il prend également en compte les ressources disponibles.
- Le *modèle de processus opérationnel* montre le chemin suivi par un processus depuis son début jusqu'à sa fin, en passant par toutes les activités qui le composent. Il intègre de plus la notion d'acteurs en charge d'une activité précise et de ressources liées à l'accomplissement d'une activité.

Pour ce travail nous avons à disposition une version standard d'Adonis, avec un module supplémentaire permettant de créer des modèles de classes, des diagrammes de cas d'utilisation et des diagrammes de séquence UML. Nous n'avons toutefois pas utilisé ce dernier, car il n'est pas très performant en comparaison avec d'autres outils UML.

## 1.2 OSSAD

OSSAD résulte d'un projet de recherche européen mené de 1985 à 1989 dans le cadre du programme ESPRIT<sup>1</sup>. Cette méthode ouverte et non-propriétaire a pour but de gérer les problèmes organisationnels amenés par l'arrivée massive de la technologie dans les bureaux (bureautique). Elle fonctionne à deux niveaux :

- Le *modèle abstrait* permet d'exprimer les objectifs d'une organisation et la représente en termes de fonctions (par exemple marketing, finance, production) et de paquets d'information qui circulent entre ces fonctions (par exemple statistiques, contrats). Les fonctions peuvent se décomposer en autant de sous-fonctions que nécessaire pour représenter une organisation, et les fonctions non décomposées sont appelées activités.
- Le *modèle descriptif* décrit les moyens humains et les ressources technologiques d'une organisation. Il la représente en terme de procédures (manière de réaliser une activité) et des différentes opérations nécessaires à leur accomplissement, ainsi qu'en termes de rôles (qui participe à quelle activité), d'outils et de ressources. Il se compose de trois types de formalismes graphiques : les *matrices activités-rôles*, les graphes de circulation des

---

<sup>1</sup> European Strategic Program for Research in Information Technology (Ancêtre des programmes cadres ICT).

informations qui décrivent la communication entre rôles (*graphe de rôles*) et entre procédures (*graphe de procédures*), ainsi que les *graphes des opérations* d'une procédure.

Pour ce comparatif nous avons utilisé l'outil OSS@D Process Design développé par la société C-Log International<sup>1</sup>. Cet outil respecte parfaitement la méthode OSSAD et offre des fonctionnalités de création et de validation de modèles. Il propose également des outils de simulation pour évaluer l'exécution des opérations d'une procédure. Seuls quelques termes OSSAD ont été changés, ainsi les *fonctions* du modèle abstrait deviennent des *processus* ou les *graphes* deviennent des *modèles*.

### 1.3 UML

UML est un langage de notation orienté objet qui a été développé et standardisé par Rational Software<sup>2</sup> et Object Management Group<sup>3</sup>. Il a vu le jour en 1997 et est très rapidement devenu un standard de l'industrie pour spécifier, visualiser, développer et documenter des logiciels. Unified Modeling Language est né, comme son nom l'indique, du regroupement de trois techniques de modélisation objet, la méthode Booch, Object Modeling Technique (OMT) et Objectory (OOSE) Process, respectivement développées par Grady Booch, James Rumbaugh et Ivar Jacobson. Ces trois chercheurs travaillent actuellement chez Rational Software.

UML couvre les différentes phases d'un développement objet (analyse, conception et implémentation) en offrant neuf types de diagrammes :

- Le *diagramme de cas d'utilisation* représente les comportements d'un système du point de vue de l'utilisateur.
- Le *diagramme de classes* représente la structure statique d'un système sous la forme de classes et de relations et ne contient pas d'informations temporelles. Une classe est une représentation abstraite d'un ensemble d'éléments similaires.
- Le *diagramme d'objets* représente les objets et leurs relations, un objet étant un élément particulier d'une classe.
- Le *diagramme de séquence* représente les objets et leurs interactions selon une ligne temporelle.
- Le *diagramme de collaboration* représente les objets, leurs liens et leurs interactions de manière structurelle. Les diagrammes de séquence et de collaboration sont isomorphes<sup>4</sup>.
- Le *diagramme de transition d'états* exprime le comportement dynamique d'un objet en termes d'états, d'activités, de transitions et d'événements.
- Le *diagramme d'activités* décrit les flux entre activités au sein d'un système. Cela permet de représenter le déroulement d'une procédure ou d'une fonction.
- Le *diagramme de composants* montre l'implémentation physique d'un système, en termes de composants logiciels.
- Le *diagramme de déploiement* décrit la configuration des éléments de traitement à l'exécution et les composants qui leur sont rattachés.

Dans ce comparatif nous nous en tiendrons aux *diagrammes de cas d'utilisation*, de *séquence*, de *collaboration* et d'*activités* car ils sont les seuls à couvrir la modélisation de processus à proprement parler. Nous avons utilisé UMLStudio, un outil qui supporte tous les diagrammes UML ainsi que d'autres méthodes de modélisation et qui permet à ses utilisateurs de

---

<sup>1</sup> [www.c-log.com](http://www.c-log.com)

<sup>2</sup> [www.rational.com](http://www.rational.com)

<sup>3</sup> [www.omg.org](http://www.omg.org)

<sup>4</sup> Cela signifie qu'il est possible de prendre un type de diagramme et de le transformer en un autre.

développer leurs propres stéréotypes. Il intègre en outre des outils de génération de code, notamment pour Java ou C++.

## 2. Types de modèles et notation graphique


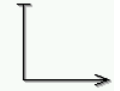


Comme annoncé en introduction, nous commençons par présenter Adonis, OSSAD et UML séparément, en détaillant les concepts ou les classes d'objets de modélisation que ces techniques proposent. Nous reprenons également la définition formelle que les créateurs de ces techniques donnent pour chacun de ces concepts. Nous n'attendons pas du lecteur qu'il retienne tous les détails, mais plutôt qu'il se forme une image générale de ces techniques, de leurs points communs et de leurs différences. Pour ce faire, nous utilisons les mêmes exemples (création et enregistrement d'une entreprise au Registre du commerce suisse) pour chaque technique, dans la mesure où il est possible de le faire car certains types modèles sont trop différents ou ne se retrouvent pas partout.

### 2.1 ADONIS

#### 2.1.1 Carte de processus

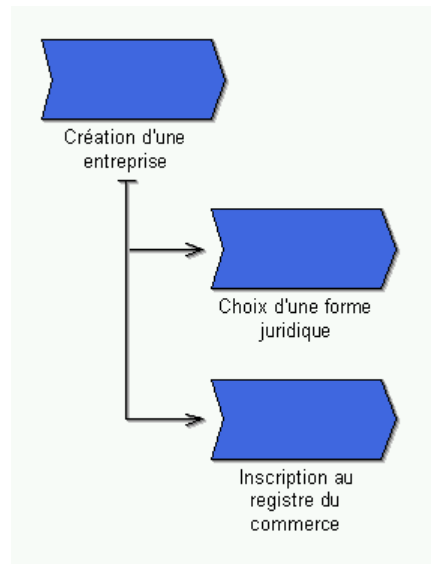
La carte de processus est un modèle Adonis qui offre une vue d'ensemble des processus opérationnels effectués dans une entreprise ou une organisation. Un processus peut être découpé en sous-processus qui figurent sur la même carte ou il peut faire référence à un autre modèle (soit une nouvelle carte des processus, soit un modèle de processus opérationnel). Les concepts utilisés dans ce type de modèle sont brièvement présentés ci-dessous.

**Table 1** : Notation pour les *Cartes de processus*

	<p>Les objets de la classe <i>Processus</i> sont utilisés uniquement dans les <i>Cartes de processus</i>. Ils peuvent faire référence à un modèle de processus opérationnel ou à une autre carte des processus.</p>
	<p>La relation <i>A le processus</i> est définie entre des objets de la classe <i>Processus</i> et sert à exprimer le découpage en sous-processus.</p>
	<p>La classe <i>Note</i> permet de stocker du texte libre dans la description graphique d'un modèle. Elle peut être utilisée dans tous les types de modèles Adonis.</p>
	<p>L'<i>Agrégation</i> permet de représenter graphiquement les liens logiques entre contenus dans les <i>Cartes de processus</i> et les <i>Modèles de processus opérationnels</i>.</p>

La figure 1 représente un exemple simplifié de création d'une entreprise. Ce processus comprend les sous-processus *Choix d'une forme juridique* et *Inscription au registre du commerce*, qui à leur tour font référence à d'autres cartes de processus ou à des modèles de processus opérationnels.







**Figure 1** : Carte de processus


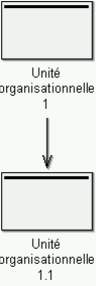

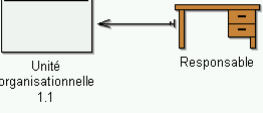

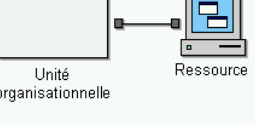
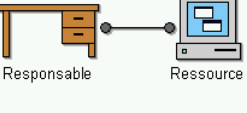
A l'image de l'exemple ci-dessus, les cartes de processus donnent à notre avis une image trop générale d'une entreprise, d'une organisation ou d'un de leurs sous-ensembles pour être réellement utiles à la modélisation. La décomposition en processus et sous-processus peut certes être utile à la compréhension d'une organisation, mais ces décompositions fonctionnelles (donc hiérarchiques) sont limitées car elles ne montrent pas facilement les liens fonctionnels entre niveaux. Elles n'apportent en effet que peu d'informations même si ce type de modèle se situe à un niveau d'abstraction élevé. Elles sont cependant utiles pour la navigation entre différents modèles au sein de l'environnement Adonis car elles constituent un point d'entrée permettant ensuite de zoomer sur les différents niveaux de représentation d'une organisation.

### 2.1.2 Modèle d'environnement de travail

Adonis est livré avec un module de modélisation des environnements de travail qui permet de représenter la structure (l'organigramme) d'une organisation. Grâce aux concepts expliqués ci-dessous, les utilisateurs peuvent modéliser aussi bien la hiérarchie d'une organisation, les personnes qui y travaillent et les rôles qu'elles y occupent, ainsi que les ressources informationnelles, matérielles et logicielles à disposition.

**Table 2** : Notation pour les Modèles d'environnement de travail

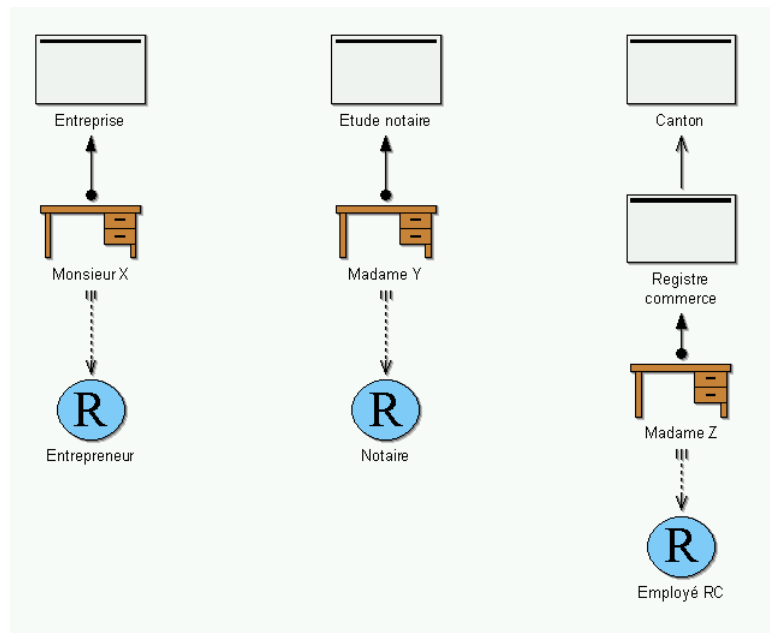
 <p>Unité organisationnelle</p>	<p>La classe <i>Unité organisationnelle</i> permet de représenter un sous-ensemble d'une organisation dans les <i>Modèles d'environnement de travail</i>. La combinaison des <i>Unités organisationnelles</i> (qui comprennent des <i>Responsables</i> et peuvent avoir un <i>Directeur</i>) permet de définir la hiérarchie d'une organisation.</p>
 <p>Responsable</p>	<p>La classe <i>Responsable</i> représente les personnes physiques dans les <i>Modèles d'environnement de travail</i>. Chaque <i>Responsable</i> peut avoir un ou plusieurs <i>Rôle(s)</i> et faire partie d'une ou plusieurs <i>Unité(s) organisationnelle(s)</i>. Un <i>Responsable</i> peut en outre être le directeur d'une ou plusieurs <i>Unité(s) organisationnelle(s)</i>.</p>

 <p>Rôle</p>	<p>La classe <i>Rôle</i> décrit le domaine d'activité d'un <i>Responsable</i>.</p>
	<p>La relation <i>Est le supérieur de</i> relie deux objets de la classe <i>Unité organisationnelle</i>. Une <i>Unité organisationnelle</i> peut être hiérarchiquement inférieure à une ou plusieurs <i>Unité(s) organisationnelle(s)</i> et une <i>Unité organisationnelle</i> peut être hiérarchiquement supérieure à une ou plusieurs <i>Unité(s) organisationnelle(s)</i>.</p>
	<p>La relation <i>Fait partie de</i> relie un objet de la classe <i>Responsable</i> à un objet de la classe <i>Unité organisationnelle</i>. Plusieurs <i>Responsables</i> peuvent faire partie d'une <i>Unité organisationnelle</i> et un <i>Responsable</i> peut faire partie de plusieurs <i>Unités organisationnelles</i>.</p>
	<p>La relation <i>Est le directeur de</i> relie un objet de la classe <i>Responsable</i> à un objet de la classe <i>Unité organisationnelle</i>. Un <i>Responsable</i> peut gérer plusieurs <i>Unités organisationnelles</i> et une <i>Unité organisationnelle</i> peut avoir plusieurs <i>Directeurs</i>.</p>
	<p>Les <i>Rôles</i> sont assignés aux <i>Responsables</i> par l'intermédiaire de la relation <i>A le rôle</i>.</p>
	<p>La relation <i>A la ressource</i> relie un objet de la classe <i>Unité organisationnelle</i> à un objet de la classe <i>Ressource</i><sup>1</sup>. Une <i>Unité organisationnelle</i> peut avoir une ou plusieurs <i>Ressource(s)</i> et une <i>Ressource</i> peut être utilisée par une ou plusieurs <i>Unité(s) organisationnelle(s)</i>.</p>
	<p>La relation <i>Utilise la ressource</i> relie un objet de la classe <i>Responsable</i> à un objet de la classe <i>Ressource</i><sup>1</sup>. Un <i>Responsable</i> peut utiliser une ou plusieurs <i>Ressource(s)</i> et une <i>Ressource</i> peut être utilisée par un ou plusieurs <i>Responsable(s)</i>.</p>

La figure 2 montre de manière générale les différentes parties impliquées dans la création d'une nouvelle entreprise, sous la forme d'un environnement de travail comprenant des unités organisationnelles, des responsables et des rôles. Une première unité organisationnelle est constituée par l'entreprise sur le point d'être créée et celle-ci comprend au moins un employé

<sup>1</sup> Les différentes classes de ressources sont définies au point 2.1.3

qui est responsable d'inscrire l'entreprise auprès du registre du commerce. A noter qu'il peut s'agir d'une seule et même personne qui décide de créer sa propre entreprise, représentée selon plusieurs points de vue. Cette différenciation est fort utile dans le cas de grandes organisations où plusieurs personnes peuvent tenir un rôle et où plusieurs rôles peuvent être joués par la même personne. Une étude de notaire forme la deuxième unité organisationnelle où un notaire aura le rôle d'officialiser la création d'une société commerciale par le biais d'un acte notarié. La nouvelle entreprise devra ensuite être enregistrée auprès du registre du commerce où un employé effectuera la transaction.




**Figure 2** : Modèle d'environnement de travail

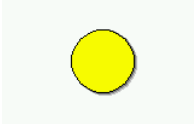

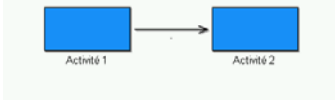




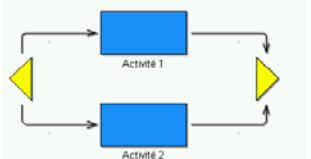


Les modèles d'environnement de travail permettent selon nous de décrire de manière exhaustive la structure d'une organisation et les ressources dont elle dispose. De plus Adonis permet de mettre en relation de manière formelle ces modèles et les modèles de processus opérationnel, ce qui permet de les valider et d'assurer ainsi une représentation «multidimensionnelle» de la réalité.


### 2.1.3 Modèle de processus opérationnel

Le *modèle de processus opérationnel* suit pas à pas le chemin d'exécution d'un processus, en détaillant séquentiellement toutes les activités qui le composent et en montrant ses différentes ramifications. Il intègre de plus la notion de responsabilité pour une activité précise et de ressources liées à l'accomplissement d'une activité.

**Table 3** : Notation pour les *Modèles de processus opérationnel*

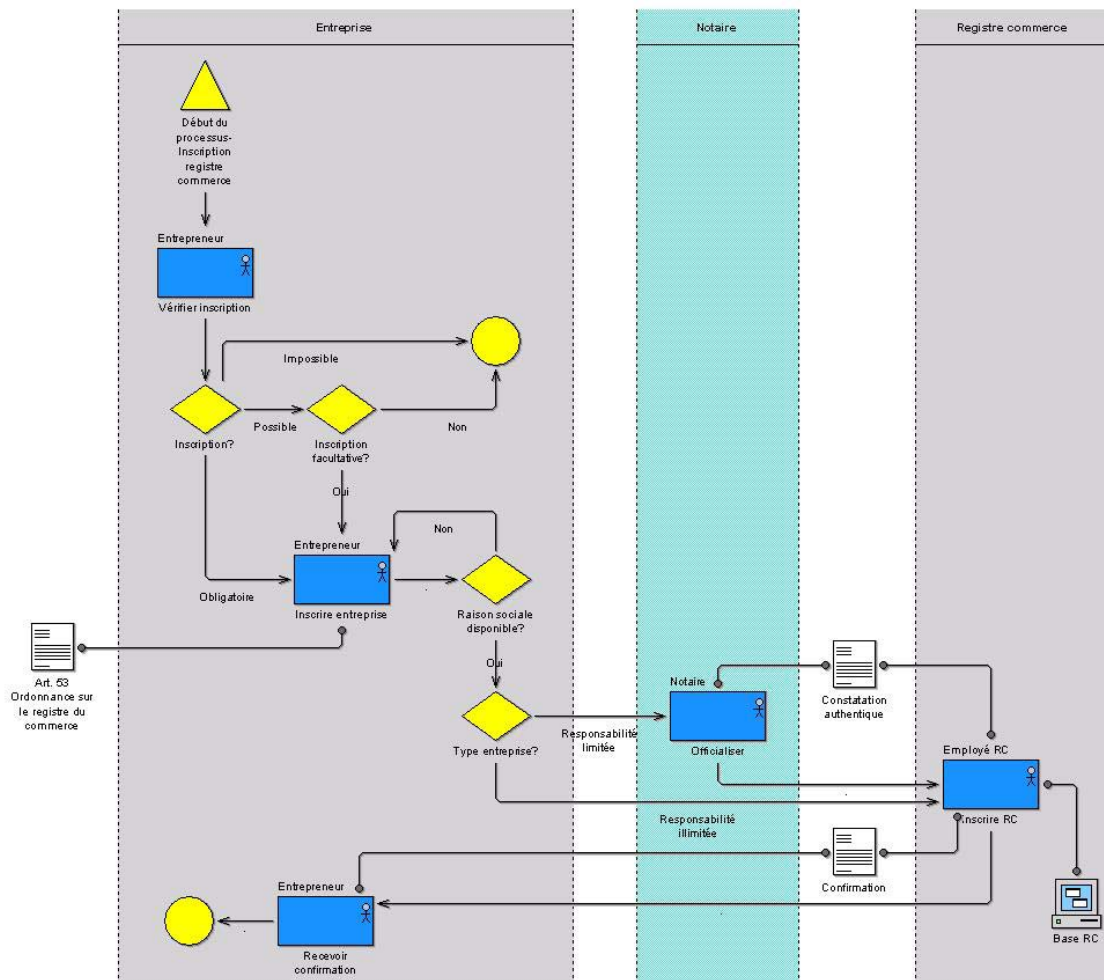
	<p>Chaque <i>Modèle de processus opérationnel</i> doit contenir exactement un objet <i>Début du processus</i> qui représente le début du processus opérationnel. Un objet de la classe <i>Début du processus</i> n'a aucun prédécesseur et a au maximum un successeur.</p>
---	--

	<p>Un objet de la classe <i>Fin</i> marque la fin du chemin d'un processus opérationnel et il peut en exister plusieurs dans un <i>Modèle de processus opérationnel</i>. L'objet <i>Fin</i> a au moins un prédécesseur et n'a pas de successeur.</p>
	<p>La classe <i>Activité</i> décrit une tâche à exécuter. Un objet de type <i>Activité</i> a un prédécesseur au moins et un successeur au maximum. Il est possible d'assigner une ou plusieurs <i>Variable(s)</i> à une <i>Activité</i>.</p>
	<p>La relation <i>Successeur</i> caractérise l'ordre de déroulement des activités et des décisions. Les utilisateurs peuvent y ajouter des conditions ou des probabilités de transition après une <i>Décision</i> ou un <i>Parallélisme</i>.</p>
	<p>Un objet de classe <i>Appel d'un processus</i> permet d'appeler d'autres processus. Il est utilisé lorsqu'un sous-processus est appelé plusieurs fois dans un modèle ou pour structurer un processus opérationnel de façon plus claire. L'objet <i>Appel d'un processus</i> doit avoir un prédécesseur au moins et un successeur maximum.</p>
	<p>Un objet de classe <i>Décision</i> permet de créer des ramifications. Ses conditions de succession doivent être définies de manière à ce qu'une seule relation de succession soit possible.</p>
	<p>Un objet de classe <i>Parallélisme</i> permet d'exécuter simultanément plusieurs chemins. Il a au moins un prédécesseur et un successeur.</p>
	<p>Un objet de classe <i>Unification</i> permet de réunir des chemins parallèles. Il doit avoir exactement autant de prédécesseurs que l'objet <i>Parallélisme</i> correspondant a de successeurs et ne doit pas avoir plus d'un successeur.</p>
	<p>Grâce aux objets <i>Parallélisme</i> et <i>Unification</i>, il est possible de traiter simultanément des parties indépendantes d'un <i>Modèle de processus opérationnel</i>.</p>
	<p>Les objets de classe <i>Swimlane</i> permettent de regrouper les activités effectuées par des classes ou des entités distinctes.</p>
	<p>La relation <i>Utilise</i> va d'un objet de la classe <i>Activité</i> à un objet de la classe <i>Ressource</i>. Elle permet d'assigner les ressources nécessaires à l'exécution d'une activité.</p>

	<p>Les objets de la classe <i>Ressource</i> décrivent les ressources nécessaires à l'exécution d'une activité et peuvent être représentés sous les différentes formes présentées ci-contre. Un objet <i>Ressource</i> dans un <i>Modèle de processus opérationnel</i> correspond exactement à l'objet du même nom défini dans un <i>Modèle d'environnement de travail</i><sup>1</sup>. Ces objets <i>Ressources</i> sont notamment pris en compte dans les algorithmes de simulation qu'offre Adonis.</p>
	<p>Les objets de classe <i>Ressource mobile</i> ont exactement la même utilité que les ressources présentées ci-dessus, si ce n'est que, comme leur nom l'indique, ils sont mobiles et circulent entre les différentes <i>Unités organisationnelles</i> et <i>Responsables</i> impliqués dans le déroulement d'un processus. Ils peuvent également être représentés sous différentes formes.</p>
	<p>Les <i>Variables</i> sont utilisées pour définir les conditions de transition des relations <i>Successeur</i> après une <i>Décision</i> ou un <i>Parallélisme</i>. L'objet <i>Variable</i> n'a ni prédécesseur ni successeur et il doit être en relation avec au moins un objet de la classe <i>Assignment de variable</i>.</p>
	<p>Un objet de classe <i>Assignment de variable</i> sert à assigner une certaine valeur à une variable, en utilisant une fonction de distribution constante ou discrète. Il n'a ni prédécesseur ni successeur.</p>
	<p>Une relation <i>Assigne</i> est définie entre un objet de la classe <i>Assignment de variable</i> et un objet de la classe <i>Début du processus</i> ou <i>Activité</i>. Elle peut être appelée dans plusieurs <i>Décisions</i> successives ou <i>Parallélismes</i>. La relation <i>Assigne une variable</i> va d'un objet de la classe <i>Assignment de variable</i> vers un objet de la classe <i>Variable</i>. Elle assigne une valeur constante ou discrète à la variable.</p>

<sup>1</sup> Voir point 2.1.2

La figure 3 détaille processus d'inscription d'une entreprise au registre du commerce en Suisse. Les swimlanes indiquent les différents acteurs impliqués et le chemin d'exécution du processus est clairement montré en fonction des conditions en vigueur. La circulation de l'information et les ressources disponibles sont également représentées. Les classes *Activités* intègrent une information supplémentaire par rapport à ce qui est expliqué ci-dessus : l'icône d'un petit personnage est ajoutée dans le coin supérieur droit et elle indique que des *Rôles* ou des *Responsables* ont été assignés à une *Activité* donnée.



**Figure 3 :** Modèle de processus opérationnel



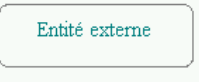


Les modèles de processus opérationnel permettent de décrire exactement et de manière formalisée le chemin d'exécution d'un processus, avec une validation immédiate par l'outil Adonis. Un des atouts de ce type de modèle est à notre sens la combinaison des swimlanes et des rôles (ou responsables), ce qui permet de dissocier les acteurs du monde réel et les ensembles de responsabilités définies à un niveau abstrait au sein de l'organisation. Les ressources nécessaires sont aussi représentées, avec une notation relativement riche qui permet de les catégoriser. Bien que les modèles de processus puissent s'imbriquer et permettre des représentations décomposées, ils ne constituent qu'un seul niveau d'abstraction. Cela peut entraîner une modélisation très opérationnelle de la réalité et empêcher une vision globale des objectifs poursuivis par une organisation ou un des ses sous-systèmes. Cette vision globale est par ailleurs relativement mal définie par les cartes de processus comme nous l'avons vu au point 2.1.1.

## 2.2 OSSAD

### 2.2.1 Modèle abstrait

Le modèle abstrait d'OSSAD s'intéresse aux objectifs d'une organisation, indépendamment des moyens et des ressources mis en œuvre pour les réaliser. Il offre plusieurs niveaux de visualisation en permettant de décomposer un processus en sous-processus et en activités grâce au « zoomage ». Il est donc relativement invariant et permet de représenter à un niveau conceptuel les processus d'une organisation, son environnement et les paquets d'information qui circulent. Les concepts utilisés pour créer un modèle abstrait sont présentés ci-après.

**Table 4** : Notation pour les *Modèles abstraits*

	<p>Un <i>Processus</i> (ou fonction) est un sous-ensemble d'une organisation qui poursuit des objectifs homogènes. L'organisation est ainsi représentée par le <i>Modèle abstrait</i>.</p>
	<p>Un <i>Processus</i> peut être formé de plusieurs sous-processus (ou sous-fonctions) : il est alors représenté par un double rectangle et l'utilisateur a la possibilité de zoomer sur un sous-ensemble du <i>Modèle abstrait</i><sup>1</sup>. Les <i>Sous-processus</i> peuvent à leur tour être décomposés en <i>Activités</i>, qui ont un objectif principal et qui constituent le niveau d'analyse le plus fin du <i>Modèle abstrait</i>.</p>
	<p>Une <i>Entité externe</i> (ou processus externe) n'appartient pas à l'organisation étudiée mais fait partie de son environnement. Une <i>Entité externe</i> est représentée dans un <i>Modèle abstrait</i> pour illustrer la circulation des <i>Paquets</i> d'information entre une organisation et son environnement, mais elle ne peut pas faire l'objet d'une décomposition.</p>
	<p>Un <i>Paquet</i> représente un ensemble d'information circulant entre <i>Processus</i> (et/ou des <i>Entités externes</i>). Il s'agit de l'unique lien possible entre ces derniers et il doit être émis par un et un seul <i>Processus</i>. Il peut par contre être reçu par plusieurs <i>Processus</i> (au minimum un), y compris par son émetteur.</p>
	<p>Les liens entre <i>Processus</i> et <i>Paquets</i> sont représentés par une flèche unidirectionnelle.</p>

La figure 4 illustre de manière très générale et relativement simplifiée l'environnement du processus qui mène à la création d'une nouvelle entreprise. Un entrepreneur constitue un dossier de création de son entreprise puis effectue un certain nombre d'activités (ou les fait effectuer par un mandataire qui se charge de ce processus) menant à la création effective de sa société, avant de se mettre en relation avec différentes administrations pour l'officialiser afin de pouvoir démarrer réellement son activité commerciale.

<sup>1</sup> Voir figures 4 et 5

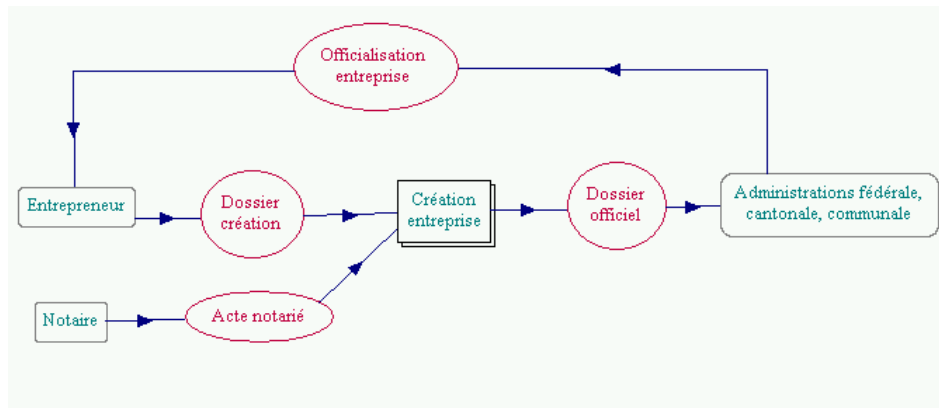


Figure 4 : Modèle abstrait

La figure 5 montre de manière plus détaillée quelles sont les activités que l'entrepreneur doit mener à bien pour créer son entreprise: choix d'une forme juridique et constitution éventuelle d'une société, obtention d'un numéro de TVA et inscription au registre du commerce si nécessaire. Elle fait de plus intervenir des entités externes plus détaillées que le modèle présenté à la figure 4 qui montrait «l'administration» de manière très abstraite.

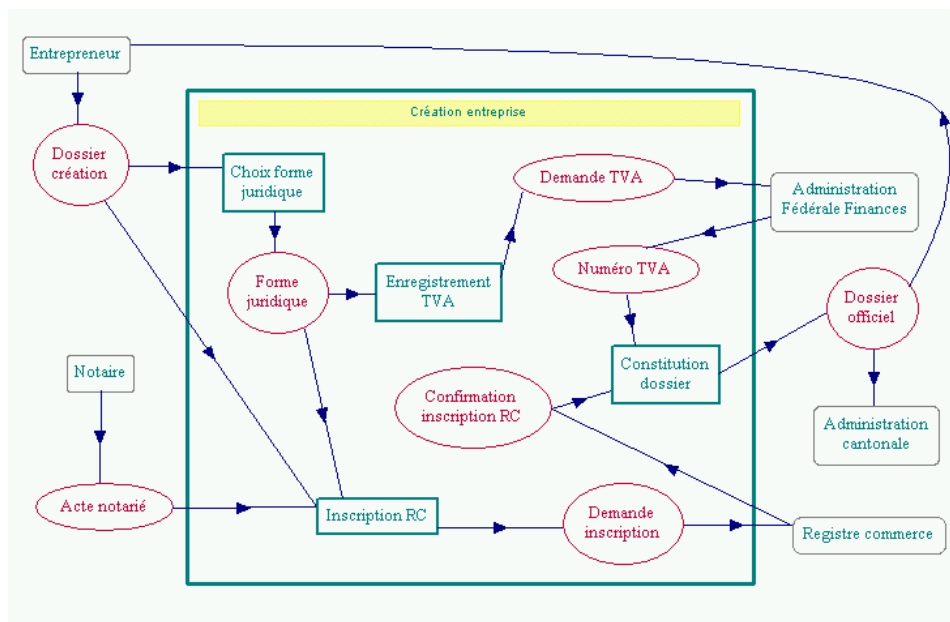


Figure 5 : Modèle abstrait zoomé

Le modèle abstrait est, comme annoncé en introduction, une représentation relativement invariante d'une organisation. Cela s'avère très utile dans un environnement où les technologies et les opérations quotidiennes peuvent évoluer rapidement, car les objectifs fondamentaux d'une organisation sont clairement formulés et ses missions formalisées, fournissant ainsi une base solide pour les personnes en charge de gérer cette évolution. Ce modèle abstrait est ensuite complété par les modèles descriptifs présentés aux points suivants : modèle de rôles, de procédures, d'opérations, d'unités organisationnelles et matrice activités-rôles.




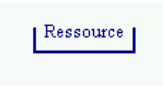

### 2.2.2 Modèle de rôles

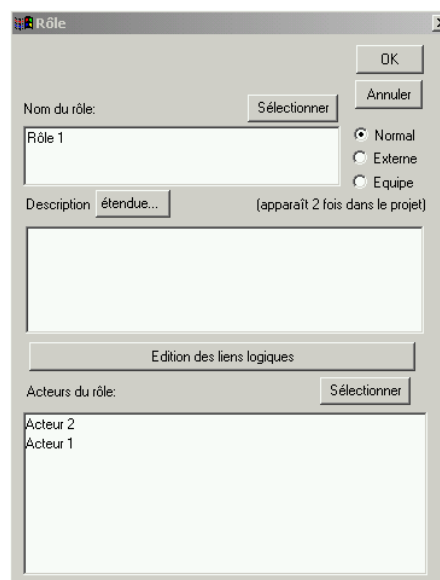
Un modèle de rôles décrit la circulation des informations dans une organisation et entre celle-ci et son environnement, en intégrant la notion de rôles, c'est-à-dire des ensembles de



responsabilités confiés aux employés ou aux personnes extérieures en contact avec l'organisation. Ces informations (ou ressources en information) sont reprises dans les modèles de procédures présentés plus loin. Les différents concepts permettant de construire un tel modèle sont listés dans la table 5.

**Table 5** : Notation pour les *Modèles de rôles*

	Un <i>Rôle</i> est un ensemble de responsabilités confiées à un ou plusieurs acteurs <sup>1</sup> dans le cadre du déroulement d'une procédure <sup>2</sup> telle que spécifiée dans un <i>Modèle de procédures</i> .
	Un <i>Rôle externe</i> est un ensemble de responsabilités appartenant à un ou plusieurs acteurs extérieurs à l'organisation, par exemple des clients ou des fournisseurs.
	Une <i>Equipe</i> regroupe plusieurs <i>Rôles</i> poursuivant Des objectifs homogènes.
	Une <i>Ressource</i> est un ensemble d'informations regroupées sur un support physique. Elle ne peut être émise que par un et un seul <i>Rôle</i> et utilisée par un et un seul <i>Rôle</i> . Les <i>Paquets</i> d'information du <i>Modèle abstrait</i> sont constitués d'une ou plusieurs <i>Ressource(s)</i> .
	Le lien entre un <i>Rôle</i> et une <i>Ressource</i> qu'il produit est montré par une flèche unidirectionnelle. Ce lien est de type <i>Emission</i> . Le lien entre une <i>Ressource</i> et le <i>Rôle</i> qui l'utilise est montré par une flèche unidirectionnelle. Ce lien peut être de type <i>Consultation</i> ou <i>Modification</i> .



**Figure 6** : Définition des acteurs jouant un *Rôle* particulier

<sup>1</sup> Voir figure 6

<sup>2</sup> Voir point 2.2.3

Comme expliqué plus haut, un *Rôle* est un ensemble de responsabilités confié à un ou plusieurs acteurs, un *Acteur* étant une personne physique, qui peut jouer un ou plusieurs rôles. Cette dépersonnalisation, autrement dit la séparation entre les personnes et les rôles qu'elles jouent au sein d'une organisation, permet de se concentrer uniquement sur les activités et les procédures à accomplir. OSS@d Process Design offre la possibilité de lier les acteurs définis dans les modèles d'unités organisationnelles<sup>1</sup> et les rôles grâce à une boîte de dialogue telle que montrée à la figure 6.

La figure 7 montre les rôles qui entrent en ligne de compte lors de l'inscription d'une entreprise au registre du commerce. Un employé de l'administration qui a le rôle de gérer le registre reçoit une demande d'inscription ainsi que des documents reprenant la raison sociale exacte et la forme de l'entreprise à enregistrer. Ces documents ont été produits sous la responsabilité de l'entrepreneur qui doit également demander un acte notarié s'il désire enregistrer certains types de sociétés. L'employé du registre du commerce effectue ensuite l'inscription (ou la refuse si elle n'est pas possible) et retourne notification d'inscription (ou de refus).

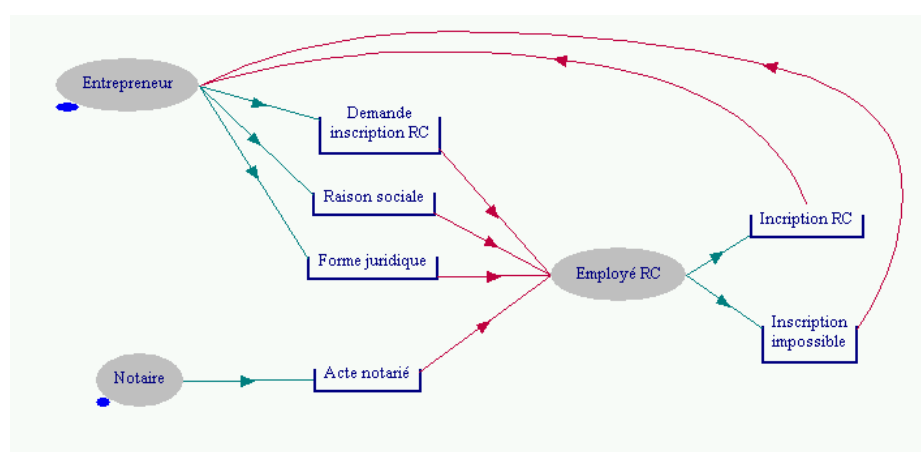


Figure 7 : Modèle de rôles

Le modèle de rôles, en parallèle avec le modèle de procédures qui regroupe les mêmes ressources en information et qui fait le lien avec le niveau abstrait<sup>2</sup>, constitue le cœur du modèle descriptif. Il montre les ressources qui circulent entre rôles et constitue une vision plus détaillée que celle du modèle abstrait car à un paquet correspondent en général plusieurs ressources en information. Mais le lien entre descriptif et abstrait est véritablement fait par le modèle de procédures.

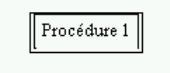
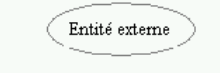


### 2.2.3 Modèle de procédures

A l'aide de la notation définie dans la table 6, le modèle de procédures permet de détailler les activités définies au niveau le plus fin du modèle abstrait : une procédure correspond à une activité. Il montre également quelles ressources en information sont nécessaires à l'accomplissement d'une activité et comment ces ressources circulent, faisant ainsi le lien avec les paquets d'information du modèle abstrait. Ce type de modèle demeure toutefois une description statique des échanges d'informations et ne s'intéresse pas à leur déroulement chronologique. Les aspects dynamiques des flux d'informations sont définis dans le modèle d'opérations associé à une procédure.

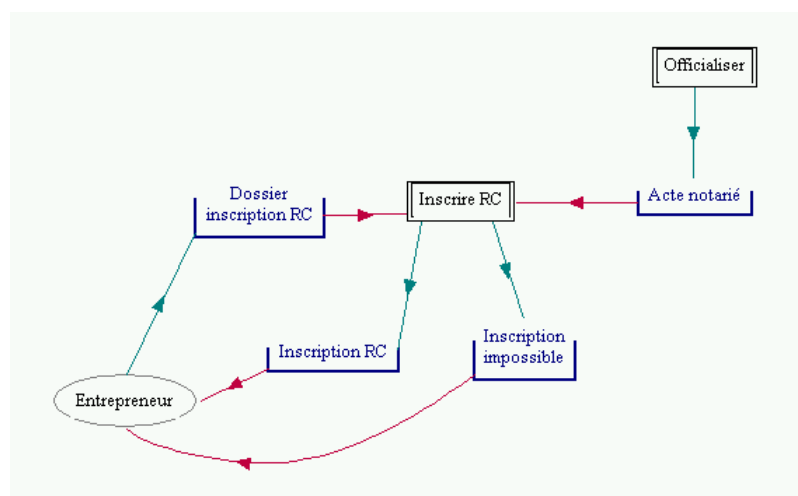
<sup>1</sup> Voir point 2.2.6

<sup>2</sup> Voir point 2.2.3

**Table 6** : Notation pour les *Modèles de procédures*

	<p>Une <i>Procédure</i> représente la manière spécifique d'accomplir une activité et elle correspond à une <i>Activité</i> définie dans le <i>Modèle abstrait</i>.</p>
	<p>Un <i>Rôle externe</i> est un ensemble de responsabilités appartenant à un ou plusieurs acteurs extérieurs à l'organisation.</p>
	<p>Une <i>Ressource</i> est un ensemble d'informations regroupées sur un support physique. Elle ne peut être émise que par une et une seule <i>Procédure</i> (ou <i>Entité externe</i>) et utilisée par une et une seule <i>Procédure</i> (ou <i>Entité externe</i>). Les <i>Ressources</i> du <i>Modèle de rôles</i> et celles du <i>Modèle de procédure</i> correspondent car ce sont en général les mêmes informations qui circulent. Les <i>Paquets</i> d'information du <i>Modèle abstrait</i> sont constitués d'une ou plusieurs <i>Ressource(s)</i>.</p>
	<p>Le lien entre une <i>Procédure</i> et une <i>Ressource</i> qu'elle produit est montré par une flèche unidirectionnelle. Ce lien est de type <i>Emission</i>. Le lien entre une <i>Ressource</i> et la <i>Procédure</i> qui l'utilise est montré par une flèche unidirectionnelle. Ce lien peut être de type <i>Consultation</i> ou <i>Modification</i>.</p>

La figure 8 montre les différentes procédures liées à l'inscription d'une entreprise au registre du commerce et les ressources en information nécessaires pour l'accomplissement de cette activité. Elle montre ainsi qu'il est nécessaire de préparer un dossier d'inscription, d'officialiser la société par un acte notarié et de procéder à l'inscription à proprement parler qui débouchera sur une notification d'inscription, positive ou négative. Elle n'intègre cependant pas de notion de temps et ne spécifie par exemple pas qu'il faut préparer le dossier avant de soumettre l'inscription.

**Figure 8** : Modèle de procédures


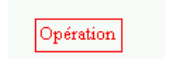
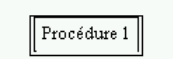

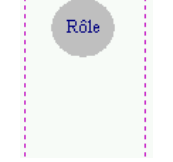


Le modèle de procédures constitue l'articulation entre les activités du niveau abstrait et la description détaillée de leur réalisation. Il permet aussi de mettre en correspondance les

ressources en information et les paquets d'information. Ce type de modèle assure donc le passage du niveau abstrait (stratégique) au niveau descriptif (opérationnel) et constitue un niveau de modélisation intermédiaire fort utile.

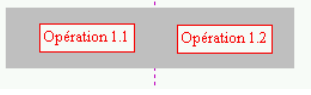
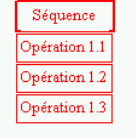
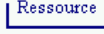

## 2.2.4 Modèle d'opérations

Le déroulement chronologique d'une procédure se décrit au moyen d'un modèle d'opérations, qui constitue le niveau de détail le plus précis de la méthode OSSAD. Une procédure est ainsi découpée en opérations «élémentaires» qui sont accomplies par un seul acteur en une seule fois. Ce type de modèle reprend par ailleurs les rôles définis dans le modèle de rôles et leur associe les opérations dont ils ont la responsabilité grâce à des colonnes similaires au concept des swimlanes d'Adonis. Il permet également de définir des chemins d'exécution parallèles ou conditionnels. La table 7 présente les symboles utilisés pour créer ces modèles.

**Table 7** : Notation pour les *Modèles d'opérations*

	<p>L'<i>Etat</i> ou l'<i>Evénement</i> est une situation particulière dans le déroulement d'une procédure. Cela permet d'illustrer un événement temporel tel que le début ou la fin d'une procédure<sup>1</sup>.</p>
	<p>Une <i>Opération</i> est une action élémentaire d'une procédure. Elle est accomplie par un et un seul acteur.</p>
	<p>Une <i>Procédure</i> représente la manière spécifique d'accomplir une activité et elle correspond à une <i>Activité</i> définie dans le <i>Modèle abstrait</i>.</p>
	<p>L'enchaînement temporel des opérations et des procédures est montré par une flèche unidirectionnelle. Les liens sont de type <i>Déclenchement</i> s'ils joignent des opérations ou des procédures, ou de type <i>Emission</i> ou <i>Réception</i> s'ils illustrent la transmission de ressources d'informations.</p>
	<p>Les <i>Rôles</i> intervenant dans une procédure sont représentés par des colonnes. Les opérations effectuées par un rôle sont placées dans la colonne correspondante. L'ensemble des opérations d'un rôle pour une procédure est appelé <i>Tâche</i>.</p>
	<p>Une <i>Pré-condition</i> est placée avant une opération et elle conditionne l'exécution de cette opération. Une <i>Pré-condition</i> peut être de type <i>Et</i> ou <i>Ou</i>, mais par convention il est courant de ne représenter que les pré-conditions de type <i>Et</i>.</p>
	<p>Une <i>Post-condition</i> est placée après une opération et elle conditionne l'exécution de la suite des opérations. Une <i>Post-condition</i> peut être de type <i>Et</i> ou <i>Ou</i>, mais par convention il est courant de ne représenter que les post-conditions de type <i>Ou</i>.</p>

<sup>1</sup> Le début et la fin d'une procédure sont en général définis implicitement dans OSSAD, mais dans certains cas il peut être utile de les marquer de façon explicite.

	<p>Une <i>Opération parallèle</i> est marquée par un rectangle gris et regroupe deux opérations effectuées simultanément. Elle peut également être représentée par une flèche bidirectionnelle.</p>
	<p>Une <i>Séquence</i> est un ensemble d'opérations effectuées à la suite par un seul acteur jouant le même rôle. Les opérations constituant la séquence sont représentées en dessous de cette dernière mais elles peuvent être masquées.</p>
	<p>Une <i>Ressource</i> est un ensemble d'informations regroupées sur un support physique.</p>
	<p>Un <i>Outil</i> représente tout moyen technique permettant la réalisation d'une opération. Il peut s'agir de matériel, de logiciel, de mobilier, etc.</p>

La figure 9 reprend le même chemin d'exécution de la procédure d'inscription d'une nouvelle entreprise que celui défini avec Adonis à la figure 3. Mis à part quelques différences au niveau de la représentation des rôles et des symboles de notation graphique, les modèles Adonis et OSSAD sont similaires. OSSAD a cependant l'avantage d'offrir une couche d'abstraction supplémentaire avec ses modèles de rôles et de procédures, alors qu'Adonis présente la réalité de manière relativement «plate».

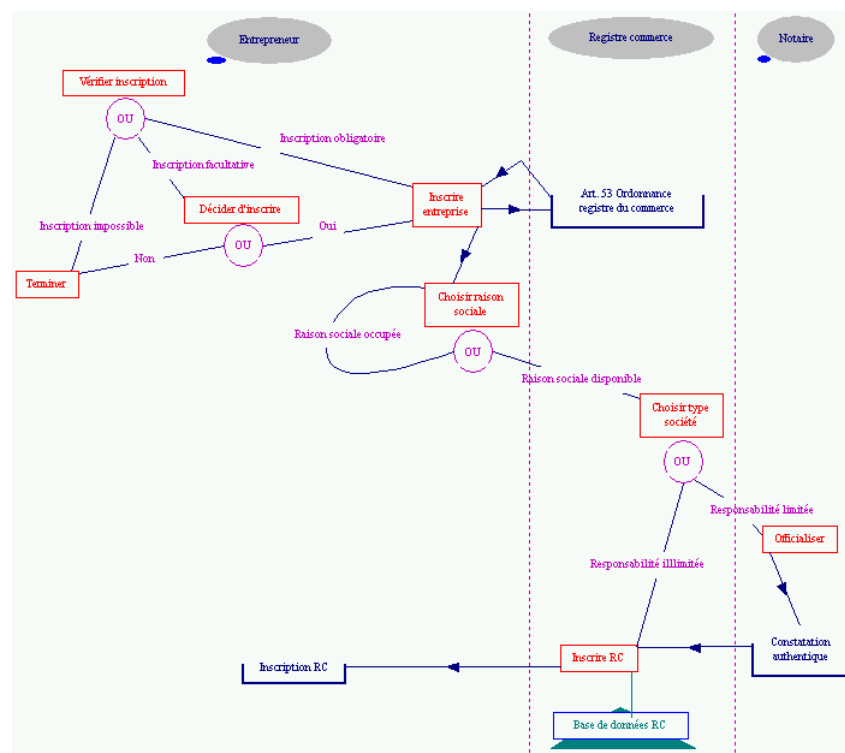


Figure 9 : Modèle d'opérations

### 2.2.5 Matrice activités-rôles

Une matrice activités-rôles se présente sous la forme d'un tableau dont chaque ligne correspond à une activité du modèle abstrait et dont chaque colonne représente un rôle du

modèle descriptif. Une croix placée dans une cellule montre la correspondance formelle entre un rôle et une activité et constitue une tâche, c'est-à-dire l'ensemble des opérations effectuées par un rôle. La matrice activités-rôles permet de lier le niveau abstrait et le niveau descriptif en montrant qui fait quoi.

L'exemple de la figure 10 montre que l'entrepreneur et le notaire sont impliqués dans le choix et la formalisation d'une forme juridique pour la nouvelle entreprise, que l'entrepreneur effectue toutes les opérations liées à la création de l'entreprise et qu'il collaborera avec un employé du registre du commerce pour effectuer l'inscription de sa société.

Création d'une nouvelle entreprise			
	Employé RC	Entrepreneur	Notaire
Choix forme juridique	.	X	X
Enregistrement TVA	.	X	.
Inscription RC	X	X	.
Constitution dossier	.	X	.

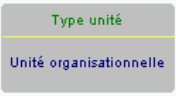

**Figure 10** : Matrice activités-rôles


Cette matrice constitue une représentation de la réalité fort originale qui ne se retrouve que dans la méthode OSSAD. Elle offre un point de vue croisé sur les activités et les rôles qui en sont responsables, ce qui permet de mettre en correspondance les différents modèles descriptifs et le modèle abstrait et qui empêche justement que les différents niveaux de modélisation soient déconnectés.

### 2.2.6 Modèle d'unités organisationnelles

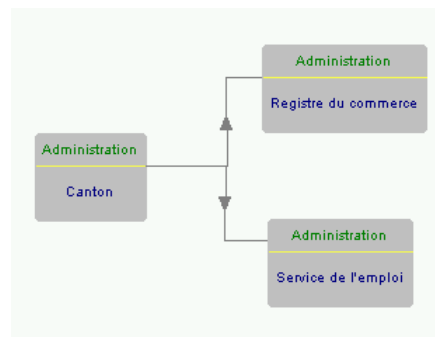
Les modèles d'unités organisationnelles permettent de montrer la structure et la hiérarchie d'une organisation, ainsi que les personnes réelles qui la composent et les postes qu'elles y occupent. Ces modèles reposent sur trois concepts simples présentés à la table 8.

**Table 8** : Notation pour les *Modèles d'unités organisationnelles*

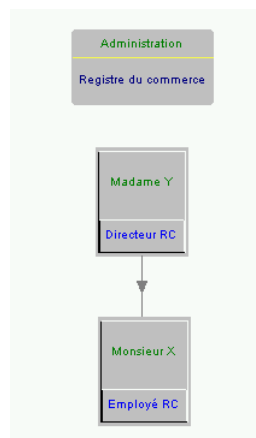
	<p>Une <i>Unité organisationnelle</i> est un découpage administratif d'une organisation. Il peut s'agir d'un département, d'un service, d'une section, etc.</p>
	<p>Un <i>Acteur</i> est une personne physique qui peut jouer différents rôles et qui occupe un <i>Poste</i>, qui peut être exprimé par son titre ou par le type d'activités qu'un acteur effectue dans l'organisation (par exemple informaticien ou chargé de la communication).</p>

	<p>Les liens entre <i>Unités organisationnelles</i> sont de type <i>Dépendance hiérarchique</i> et sont représentés par une flèche unidirectionnelle grise. Les liens entre <i>Acteurs</i> peuvent uniquement être de type <i>Supérieur de</i> et utilisent le même formalisme. Tous les Acteurs placés en dessous d'une Unité en font partie<sup>1</sup>.</p>
---	--

Les modèles d'unités organisationnelles de OSS@D Process Design sont légèrement différents de ceux proposés par la méthode OSS@D, même s'ils contiennent les mêmes informations : les organigrammes sont représentés de gauche à droite (figure 11) et non pas en arborescence verticale et les acteurs sont montrés dans la décomposition des unités organisationnelles (figure 12). Les graphes d'acteurs n'existent par ailleurs pas dans OSS@D Process Design, car ils ne sont pas nécessaires. En effet, les rôles attribués aux acteurs<sup>2</sup> sont repris automatiquement dans les modèles d'unités organisationnelles.



**Figure 11** : Modèle d'unités organisationnelles



**Figure 12** : Acteurs d'une unité organisationnelle

Ces modèles sont utiles à la représentation d'une organisation où les structures administratives sont couramment assez rigides. Ils permettent de lister les personnes réelles qui travaillent dans une organisation pour ensuite leur attribuer des rôles avec les modèles du même nom. Les modèles d'unités organisationnelles ne sont toutefois pas directement liés à la modélisation de processus, qui est l'objet premier de ce travail.

<sup>1</sup> Voir figure 12

<sup>2</sup> Voir figure 6



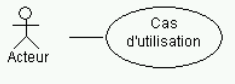

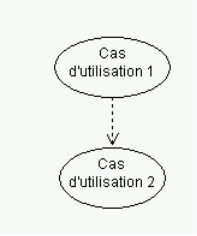
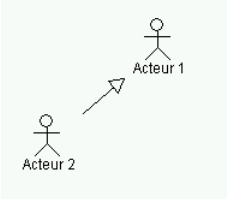
## 2.3 UML

UML est une méthode très vaste. Nous nous limitons à la présentation de trois types de modèles sur les neuf qu'elle possède : ceux les plus utiles à la modélisation de processus.

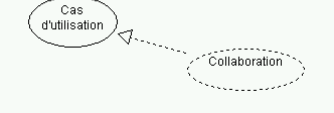
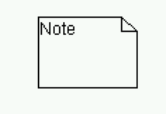
### 2.3.1 Diagramme de cas d'utilisation

Les diagrammes de cas d'utilisation ont pour but premier de décrire les interactions d'un système avec son environnement, et en particulier avec les acteurs qui le constituent, qu'ils soient des êtres humains ou des machines. Les cas d'utilisation représentent un ensemble abstrait d'actions et d'activités accomplies par un système, ce qui est appelé par les concepteurs d'UML son comportement. Ces diagrammes permettent de spécifier graphiquement ce comportement et ils peuvent être complétés par des descriptions textuelles ou des scénarios.

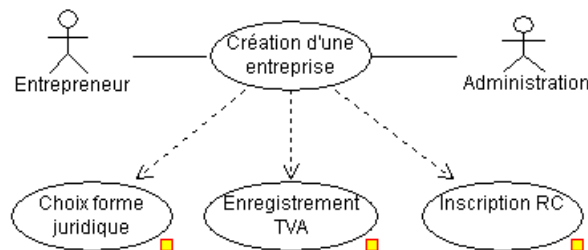
**Table 9** : Notation pour les *Diagrammes de cas d'utilisation*

	<p>Un <i>Acteur</i> représente un ensemble cohérent de rôles qui interagit avec un cas d'utilisation. Il peut s'agir d'un être humain, d'un système informatique ou de matériel spécialisé.</p>
	<p>Un <i>Cas d'utilisation</i> décrit ce qu'un système fait mais ne spécifie pas comment il le fait. Les cas d'utilisation sont souvent complétés par des scénarios textuels.</p>
	<p>Un <i>Acteur</i> peut être connecté avec un <i>Cas d'utilisation</i> uniquement par une relation d'<i>Association</i> qui se dessine par un trait plein.</p>
	<p>Le rectangle placé autour des cas d'utilisation sert à délimiter les frontières d'un système. Tout ce qui est à l'extérieur constitue l'<i>Environnement</i> de ce système.</p>
	<p>Un <i>Cas d'utilisation</i> peut inclure ou étendre un autre <i>Cas d'utilisation</i>, ce qui se montre par une flèche pointillée dont la légende dit s'il s'agit d'une <i>Inclusion</i> ou d'une <i>Extension</i>. L'inclusion s'utilise notamment pour éviter de modéliser plusieurs fois des événements identiques qui ont lieu dans plusieurs cas d'utilisation (par exemple valider un utilisateur). L'extension sert notamment à modéliser un ensemble optionnel d'événements d'un cas d'utilisation (Voir figure 13).</p>
	<p>Les <i>Acteurs</i> peuvent être définis de manière générique par le biais de stéréotypes. Les acteurs spécialisés sont des «descendants» de ces stéréotypes ou super-classes. Cela constitue une relation de <i>Généralisation</i> qui se marque par une flèche dont la pointe est vide et qui va de l'acteur spécialisé vers l'acteur générique. Un acteur <i>Employé</i> est par exemple une généralisation d'un acteur <i>Secrétaire</i>.</p>



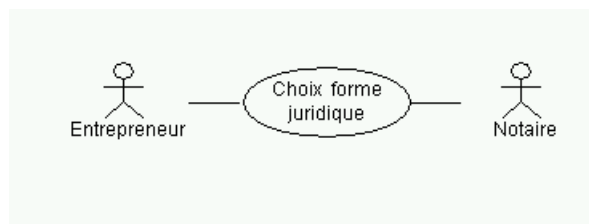
	<p>Comme mentionné plus haut, un <i>Cas d'utilisation</i> n'explique pas comment un système agit. Pour cela il existe le concept de <i>Collaboration</i> qui spécifie le fonctionnement d'un système. L'association entre une collaboration et un cas d'utilisation est appelée <i>Réalisation</i>.</p>
	<p>Une <i>Note</i> permet de stocker du texte libre dans la description graphique d'un modèle et peut être utilisée dans tous les types de modèles UML.</p>

La figure 13 montre les acteurs intervenant dans la création d'une entreprise, à un niveau d'abstraction élevé. Il est intéressant de voir que les cas d'utilisation correspondent bien aux processus définis dans le modèle abstrait d'OSSAD, mais que là où OSSAD représentait des paquets d'information circulant entre processus, UML montre les interactions entre les acteurs et les cas d'utilisation.



**Figure 13** : Diagramme de cas d'utilisation

Au même titre qu'il peut y avoir plusieurs modèles imbriqués au niveau abstrait d'OSSAD, les cas d'utilisation peuvent se décomposer en plusieurs degrés de représentation. La figure 14 précise le cas d'utilisation *Choix de la forme juridique* et montre que dans certains cas il est nécessaire d'utiliser les services d'un notaire pour créer une société commerciale.



**Figure 14** : Zoom d'un cas d'utilisation

Les cas d'utilisation permettent avant tout de spécifier et de valider les besoins des utilisateurs d'un système donné et ils reposent sur une notation très simple. Contrairement au modèle abstrait d'OSSAD, ils ne sont pas destinés à modéliser une organisation et les flux d'information qui y circulent. Cependant ils constituent selon nous un outil efficace pour modéliser des processus.

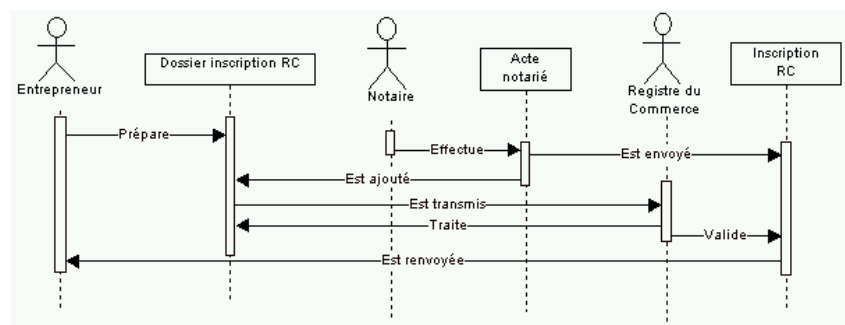
### 2.3.2 Diagramme de séquence

Les diagrammes de séquence et les diagrammes de collaboration présentés au point suivant sont appelés de manière générique diagrammes d'interactions dans UML. En effet, ils décrivent les interactions entre les objets que contient un système ou un processus. Ils permettent également de montrer les interactions d'un système avec son environnement. Les diagrammes de séquence permettent de modéliser un système de manière dynamique et ils s'attachent principalement à montrer la circulation et l'ordre chronologique des messages, autrement dit ils décrivent la circulation de l'information. Ils sont directement basés sur les scénarios définis avec les cas d'utilisation. Les diagrammes de collaboration quant à eux montrent ces interactions sous un angle structurel plutôt que temporel.

**Table 10** : Notation pour les *Diagrammes de séquence*

	<p>Les <i>Rôles</i> et les <i>Objets</i> ne représentent pas des personnes réelles ou des objets concrets. Ils représentent respectivement un ensemble de responsabilités qu'un ou des acteurs ont et un objet générique d'une classe donnée. Ils sont dessinés avec une ligne pointillée qui symbolise l'écoulement du temps (de haut en bas) et qui s'appelle <i>Ligne de vie</i>. Elle représente la durée de vie d'un objet et elle est complétée par un fin rectangle qui montre la période pendant laquelle un objet est actif.</p>
	<p>Les <i>Messages</i> échangés entre les objets peuvent être de plusieurs types : <i>Crée</i>, <i>Détruit</i>, <i>Envoie</i>, <i>Appelle</i> et <i>Retourne</i>. Ci-contre les messages <i>Appelle</i> et <i>Retourne</i> sont représentés, qui permettent respectivement d'appeler une opération d'un objet ou de retourner une valeur. L'ordre des messages est défini par leur emplacement le long de la ligne de vie.</p>

La figure 15 montre précisément l'ordre des messages échangés lors du processus d'inscription d'une entreprise au registre du commerce. Ce diagramme de séquence montre les objets *Dossier d'inscription RC*, *Acte notarié* et *Inscription au registre du commerce* et les messages qui sont échangés entre les acteurs *Entrepreneur*, *Notaire*, *Registre du commerce* et ces objets. Notons que les diagrammes de séquence ne permettent pas de définir des conditions ou des activités parallèles, qui seront définies dans les diagrammes d'activités.






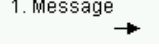
**Figure 15** : Diagramme de séquence

Les diagrammes de séquence sont en principe destinés à modéliser un système informatique et ses composants, mais nous pensons qu'il est possible d'étendre le concept de système à celui de processus car la finalité du système devient l'objectif du processus, sa valeur ajoutée. De cette manière, il devient possible de modéliser des processus également et de créer un niveau intermédiaire entre le modèle abstrait et le modèle opérationnel, en mettant l'accent sur la circulation de l'information.

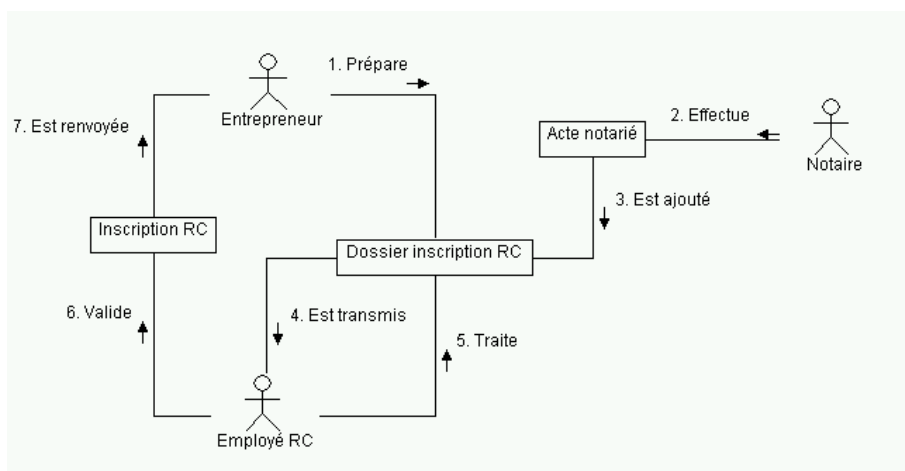
### 2.3.3 Diagramme de collaboration

Ainsi que nous l'avons expliqué au point précédent, les diagrammes d'interaction UML comprennent les diagrammes de séquence et les diagrammes de collaboration. Ces deux types de diagramme sont d'ailleurs symétriques, c'est-à-dire qu'il est possible de passer de l'un à l'autre car ils sont sémantiquement équivalents. Ils présentent néanmoins des points de vue différents : alors que le diagramme de séquence repose sur une échelle temporelle, le diagramme de collaboration se concentre sur le côté structurel des interactions d'un système et montre l'organisation des flux d'information.

**Table 11** : Notation pour les diagrammes de collaboration

 Rôle	Un <i>Rôle</i> ne représente pas une personne réelle, mais un ensemble de responsabilités qu'un ou des acteurs ont.
 Objet	Un <i>Objet</i> ne représente pas quelque chose de concret ou de réel, mais un objet générique d'une classe donnée.
	Les <i>Liens</i> entre les objets et les rôles montrent l'organisation ou la structure d'une interaction.
 1. Message	Les <i>Messages</i> échangés entre les objets ou les rôles sont ajoutés au-dessus ou à côté des liens qui les unissent. Généralement un <i>Numéro de séquence</i> leur est ajouté pour indiquer l'ordre chronologique des messages.

La figure 16 montre la circulation des informations et de leur support entre les différents acteurs impliqués dans l'enregistrement d'une entreprise au registre du commerce. Les aspects chronologiques de ce diagramme sont définis par l'ordre de numérotation des messages.





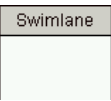

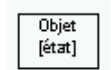


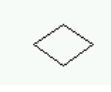
**Figure 16** : Diagramme de collaboration

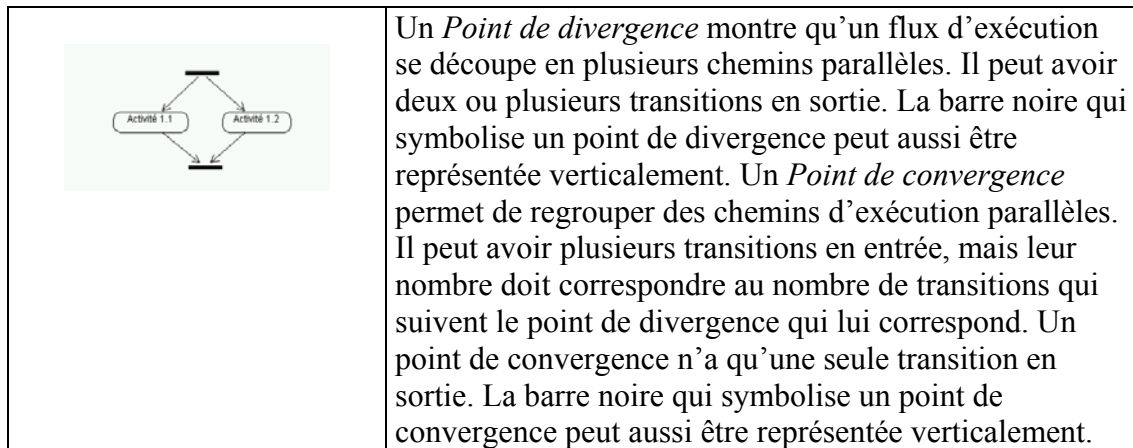
Bien que les diagrammes de collaboration d'UML soient destinés à modéliser un système informatique, nous pensons qu'ils sont clairement le pendant des modèles de rôles d'OSSAD. En effet ils décrivent également la circulation de l'information entre les acteurs d'un système, même s'ils sont légèrement différents car ils ne font pas une distinction claire entre acteurs et rôles et qu'ils permettent d'ajouter un aspect dynamique grâce à l'ordonnement des messages. Il n'en demeure pas moins que les diagrammes de collaboration jouent un rôle très important dans le travail de modélisation car ils sont le pivot entre le niveau abstrait des cas d'utilisation et le niveau détaillé décrit par les diagrammes d'activités présentés au point suivant.

### 2.3.4 Diagramme d'activité

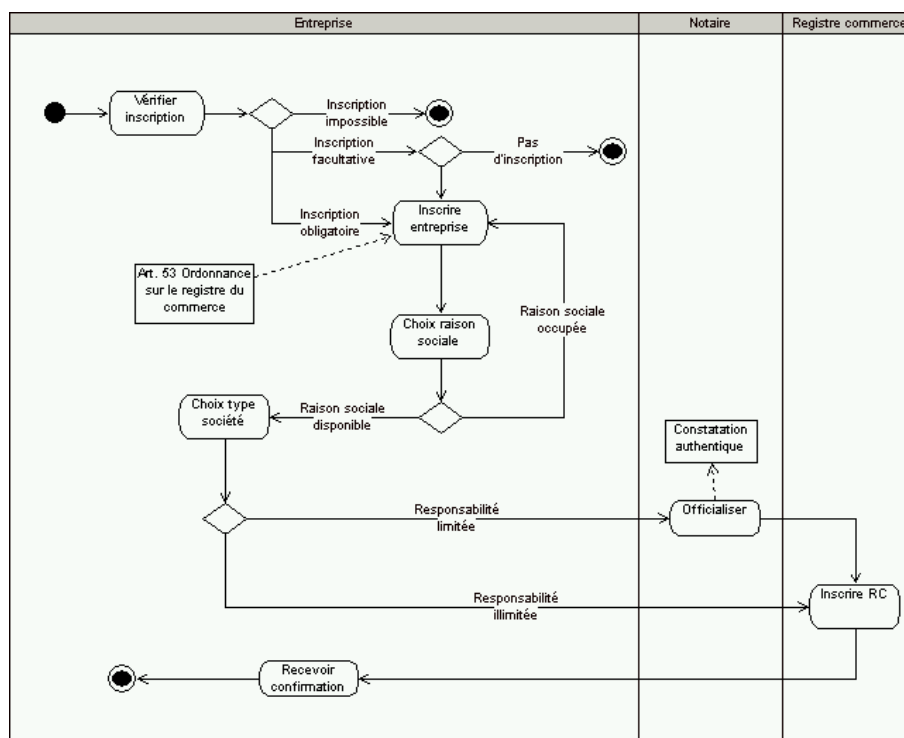
Les diagrammes d'activités montrent les flux de contrôle et les flux d'information qui circulent au sein d'un système ou d'un sous-système. Ils permettent de représenter le séquençage exact des activités et de définir des conditions d'exécution. Il est de plus possible d'attribuer des activités à des acteurs grâce aux swimlanes.

**Table 12** : Notation pour les *Diagrammes d'activités*

	L' <i>Etat initial</i> marque le début d'un <i>Diagramme d'activités</i> , c'est-à-dire le point de départ par défaut d'un flux de contrôle d'activités.
	L' <i>Etat final</i> marque la fin d'un <i>Diagramme d'activités</i> , c'est-à-dire le point où un flux de contrôle d'activités a été complètement exécuté.
	Les <i>Swimlanes</i> permettent de séparer des activités exécutées par des entités ou des acteurs différents. Elles représentent des entités ou des acteurs du monde réel, contrairement aux rôles ou aux objets des modèles d'interaction.
	Une <i>Activité</i> représente l'exécution d'actions atomiques ou d'opérations et elle cause un changement de l' <i>Etat</i> du système.
	Un <i>Objet</i> peut être nécessaire à la réalisation d'une activité et il peut circuler dans un diagramme d'activité. Il est de plus possible d'écrire l' <i>Etat</i> de l'objet entre crochets en dessous de son nom. Une relation de <i>Dépendance</i> est utilisée pour lier un objet à une ou des activités.
	Une relation de <i>Transition</i> montre le chemin d'exécution d'un flux de contrôle d'activités.
	Une relation de <i>Dépendance</i> est utilisée pour montrer la participation d'un objet au déroulement d'une activité.
	Le symbole de <i>Ramification</i> montre qu'il existe plusieurs transitions ou chemins d'exécution possibles conditionnés par une expression booléenne. Une ramification a une transition en entrée et au minimum deux transitions en sortie.



Le diagramme représenté à la figure 17 est quasiment similaire au modèle de processus opérationnel d'Adonis (figure 3) et au modèle d'opérations OSSAD (figure 9) correspondants, même si Adonis et UML utilisent le terme d'activité, alors qu'OSSAD parle d'opération. Il représente la procédure d'inscription d'une entreprise au registre du commerce de manière parfaitement équivalente.



**Figure 17** : Diagramme d'activités

Les diagrammes d'activités sont indispensables à la modélisation de processus, car ils décrivent précisément comment les processus ou sous-processus sont réalisés.

### 3. Comparaison ADONIS-OSSAD-UML

Comme nous l'avons vu au chapitre 2, les modèles d'Adonis, d'OSSAD et d'UML couvrent des champs d'application qui ne sont pas toujours similaires. Ces trois techniques reposent toutefois sur l'idée de représenter la réalité avec des points de vue et des niveaux différents, intégrant pour cela des modèles présentés en cascade et des possibilités de zoom. Afin de pouvoir les comparer, nous proposons un découpage en trois niveaux de modélisation que nous résumons en trois interrogations :

- **Quoi ?** Quels sont les objectifs de l'organisation ?
- **Qui et avec quoi ?** Quelle est la structure de l'organisation et quelles sont les ressources disponibles ?
- **Comment ?** Quel est le fonctionnement procédural de l'organisation ?

La table 13 montre la répartition que nous proposons, à savoir quels modèles répondent à quelle question. Nous expliquerons cette répartition et nous étudierons chaque niveau dans le détail aux points suivants.

**Table 13** : Matrice de comparaison

	<b>ADONIS</b>	<b>OSSAD</b>	<b>UML</b>
<b>Quoi ?</b>	Carte des processus	Modèle abstrait	Cas d'utilisation
<b>Qui et quoi ?</b>	Modèle d'environnement de travail	<div style="border: 1px solid black; border-radius: 50%; width: 100px; height: 30px; margin: 0 auto; display: flex; align-items: center; justify-content: center;">MAR</div> <ul style="list-style-type: none"> <li>- Modèle d'unités organisationnelles</li> <li>- Modèle de rôles</li> <li>- Modèle de procédures</li> </ul>	<ul style="list-style-type: none"> <li>- Diagramme de séquence</li> <li>- Diagramme de collaboration</li> </ul>
<b>Comment ?</b>	Modèle de processus opérationnel	Modèle d'opérations	Diagramme d'activités

#### 3.1 Fonctionnement de l'organisation

La description des flux de contrôle et des flux d'information est selon nous clairement identique dans Adonis, OSSAD et UML. En effet, elle repose sur des concepts communs, même si la notation est parfois différente :

- Des activités ou des opérations élémentaires qui doivent être effectuées et qui sont ordonnées de manière chronologique
- Des swimlanes qui permettent de montrer quels acteurs ou quels rôles sont responsables de ces activités ou opérations
- Des conditions et des opérations parallèles permettant de contrôler le déroulement ou la séquence de ces activités ou opérations
- Des ressources en information et des outils qui sont liés aux activités ou opérations.

Ces techniques présentent toutefois quelques différences, relativement minimes à notre avis :

- Adonis permet de différencier rôles et acteurs directement dans le modèle, alors qu'OSSAD ne traite que des rôles et qu'UML ne marque pas la différence entre rôle et acteur.
- Adonis offre un grand nombre de ressources prédéfinies qu'il n'est pas possible d'étendre alors qu'UML permet de modéliser n'importe quel type de ressources grâce au concept de stéréotype et de classes d'objets. OSSAD de son côté propose trois notations distinctes, les ressources en information, les outils et les documents.

Cette correspondance directe entre les modèles opérationnels proposés par les trois techniques n'a rien de surprenant, dans la mesure où la représentation d'une séquence d'opérations ou d'activités est d'un faible niveau d'abstraction et doit coller à la réalité. Ces modèles sont directement inspirés des ordigrammes et autres flowcharts communs à beaucoup de méthodes.

### **3.2 Structure et ressources de l'organisation**

C'est à ce niveau de représentation qu'Adonis, OSSAD et UML présentent les plus grandes différences. Ces dernières proviennent selon nous de la conception initiale de ces techniques et des champs d'application pour lesquels elles ont été prévues :

- UML est une méthode de conception de systèmes informatiques et elle ne s'intéresse de ce fait pas à la hiérarchie ou à la structure d'une organisation. Elle n'intègre donc pas directement de possibilités de modéliser de manière clairement différenciée les acteurs physiques et les rôles qu'ils ont à tenir au sein d'une organisation.
- Adonis et OSSAD sont des méthodes directement conçues pour la modélisation de processus et elles permettent de modéliser la structure d'une organisation.

Nous pensons toutefois qu'il est possible de mettre en correspondance des modèles provenant de ces trois techniques :

- Les modèles d'environnement de travail d'Adonis et les modèles d'unités organisationnelles d'OSSAD permettent de représenter les mêmes informations. UML n'offre rien de tel, mais il serait possible de contourner cette lacune grâce au concept de classes (d'acteurs collaborateurs d'une unité organisationnelle) et de généralisation. Adonis et OSSAD permettent par ailleurs d'attribuer formellement des rôles à des acteurs physiques, ce qui n'est pas prévu dans UML.
- Il n'existe aucun modèle Adonis permettant de voir la circulation d'informations entre les rôles ou les acteurs, alors qu'OSSAD et UML présentent une certaine similitude entre leur modèle de rôles et de collaboration. En effet, le premier montre la circulation de ressources d'information entre des rôles et le second les échanges de messages entre des acteurs. A noter toutefois qu'UML permet d'ajouter facultativement une numérotation correspondant à la chronologie des messages, alors que le modèle de rôles d'OSSAD ne contient pas d'information temporelle.
- Adonis n'offre pas non plus de modèle permettant de mettre en correspondance le niveau abstrait et le niveau descriptif. Là encore, nous pensons qu'OSSAD et UML ont des modèles qui présentent une certaine similitude. En effet, le modèle de procédures d'OSSAD est lié au modèle abstrait car chaque procédure représente une activité du modèle abstrait, alors que dans UML un diagramme de séquence repose sur le scénario défini pour le cas d'utilisation correspondant. Il est intéressant de constater que cette double symétrie existe également au niveau de la conception des méthodes OSSAD et UML : les modèles de rôles et de procédures OSSAD sont liés car ils sont tous deux

définis comme des modèles de circulation de l'information, alors que les modèles de collaboration et de séquence UML sont symétriques et portent le nom général de diagrammes d'interaction. Par ailleurs, la différence soulevée au point précédent est valable ici également, le diagramme de séquence UML contient une chronologie alors que le modèle de procédures n'en a pas.

- OSSAD est la seule méthode à formaliser la correspondance entre les trois niveaux que nous avons définis grâce à la matrice activités-rôles : elle permet de définir les rôles qui sont responsables d'activités données, et à une activité du niveau abstrait correspond une procédure définie au niveau opérationnel. Dans Adonis il est toutefois possible de visualiser les modèles de processus opérationnel de manière tabulaire, ce qui permet de voir quels rôles sont responsables de quelles activités (au sens Adonis, ce qui correspond donc à une opération OSSAD). Dans UML, les acteurs sont directement intégrés aux modèles, et ce déjà au niveau abstrait, contrairement à Adonis et OSSAD.

### 3.3 Objectifs de l'organisation

Les cartes de processus Adonis, les modèles abstraits d'OSSAD et les cas d'utilisation ont un but commun, celui de modéliser les objectifs d'une organisation. Ils sont cependant conçus de manière différente et ne présentent pas la même information :

- Les cartes de processus Adonis montrent uniquement des processus de manière très générale et ne contiennent guère d'informations.
- OSSAD reprend l'idée de processus, mais y ajoute le concept de paquet d'information et montre la circulation de paquets entre processus. Cette méthode intègre de plus l'idée de processus externe afin de représenter la circulation de l'information non seulement à l'intérieur d'une organisation, mais aussi entre cette dernière et son environnement.
- Les cas d'utilisation d'UML peuvent être mis directement en correspondance avec les processus d'Adonis ou d'OSSAD. Le concept d'acteurs dans UML est de plus relativement similaire à celui de processus ou d'entité externe dans OSSAD. Jusque là, nous pouvons dire qu'OSSAD et UML sont proches, mais la grande différence se situe au niveau des relations qui unissent ces processus ou cas d'utilisation. Là où OSSAD s'intéresse en premier lieu à circulation de l'information entre processus, UML spécifie de simples associations entre acteurs et cas d'utilisation et ne donne aucune précision sur le type d'informations qui circule entre eux.

### 3.4 Concepts de modélisation

Comme nous l'avons mentionné au long de ce travail, Adonis, OSSAD et UML intègre un certain nombre de concepts communs, même s'ils portent parfois des noms différents. Pour faciliter la mise en correspondance des ces concepts, nous les avons regroupé dans la table 14.

**Table 14** : Tableau de correspondance approximative des principaux concepts

	ADONIS	OSSAD	UML
Quoi ?	Processus	Processus	Cas d'utilisation
	---	Entité externe	Acteur
	Note	Annotation	Note
	Agrégation	Processus zoomé	Environnement



<b>Qui et quoi ?</b>	Unité organisationnelle	Unité organisationnelle	---
	Responsable	Acteur	Acteur
	Rôle	Rôle	Acteur
	Ressource	Ressource	Objet
<b>Comment ?</b>	Activité	Opération	Activité
	Décision	Post-condition	Ramification
	Parallélisme	Opération parallèle	Points de divergence et de convergence
	Swimlane	Rôle	Swimlane
	Ressource	Ressource / Outil	Objet

## 4. Conclusion

Dans les chapitres précédents, nous avons vu qu'au niveau opérationnel, les concepts étaient très similaires dans Adonis, OSSAD et UML. Nous pouvons donc dire qu'à ce niveau, les trois techniques peuvent être utilisées indifféremment pour modéliser un processus et que le passage d'une technique à l'autre peut se faire facilement.

Au niveau abstrait, nous relevons que le concept de processus est présent partout, mais qu'une des techniques l'utilise tel quel (Adonis), que l'autre lui ajoute des paquets d'information (OSSAD) et que la dernière y intègre des acteurs (UML). Après notre comparaison, nous pensons le niveau abstrait d'Adonis est trop général pour être réellement utile, alors qu'UML et OSSAD sont plus riches au niveau de la représentation. Nous pouvons même dire qu'OSSAD est la méthode la plus détaillée, car le concept d'entité externe permet de reprendre les acteurs définis dans UML et qu'elle est la seule à s'intéresser à la circulation de l'information.

En ce qui concerne la description structurelle d'une organisation, nous avons constaté une différence importante entre la paire Adonis et OSSAD, les méthodes de modélisation de processus à proprement parler, et UML qui est plutôt destiné à modéliser des systèmes d'information. Ainsi le choix d'une méthode dépendra du champ d'application du travail de modélisation de processus, afin de pouvoir utiliser au mieux les fonctionnalités respectives de ces trois techniques de modélisation. Nous jugeons toutefois qu'OSSAD est la méthode qui assure le mieux la liaison entre les modèles structurels et le niveau abstrait grâce à la matrice activités-rôles.

Comme annoncé en introduction, nous n'avons pas réalisé ce travail pour déterminer quelle est la meilleure technique de modélisation ou quelle est la moins bonne. Nous terminons néanmoins par quelques lignes d'appréciation sur chacune d'entre elles :

- Adonis ne couvre pas très bien la modélisation du niveau abstrait car elle n'offre pas suffisamment de concepts de description. Au niveau de la structure et des ressources, ainsi qu'au niveau opérationnel, elle est tout à fait satisfaisante, parfois même trop complète pour faciliter une vue d'ensemble susceptible de faciliter une réorganisation.
- OSSAD permet de couvrir tous les aspects de la modélisation de processus et ses différents niveaux de modèles sont fort bien articulés entre eux. Des extensions

implémentées dans le logiciel Workey™ permettent de générer automatiquement des applications de workflow.

- UML est une technique plus générique, avec les avantages et les inconvénients que cela implique : elle ne «force» pas l'utilisation de certains concepts qui peuvent s'avérer très importants dans la modélisation de processus, entraînant ainsi une perte d'information ou la création de modèles incomplets, mais elle est par contre flexible et extensible, ce qui permet à ses utilisateurs de l'adapter précisément à leurs besoins. UML permet également la génération automatique de code applicatif.

## 5. Bibliographie

Il ne s'agit pas ici d'une bibliographie traditionnelle au sens où des publications citées dans le texte seraient reprises ci-dessous. Ces titres sont plutôt donnés à titre de référence générale sur les méthodes et outils utilisés dans ce travail.

BOC (2002). *Formation de base à Adonis*. Classeur de formation reçu lors d'un cours donné pour l'IDHEAP en février 2002.

BOC (2000). *Fichiers d'aide de l'outil Adonis*. Ce logiciel est bien documenté, notamment en tout ce qui concerne les bibliothèques de classes, et cela nous a été très utile pour la description de la notation Adonis.

Booch, G., Rumbaugh, J. & Jacobson, I. (1999) *The Unified Modeling Language User Guide*. Addison-Wesley.

Chappelet, J.-L. & Snella, J.-J. (1997) *Un langage pour l'organisation : l'approche OSSAD*. Presses Polytechniques et Universitaires Romandes.

C-Log (2001) *Tutorial OSS@D Process Design*. Document fourni avec le logiciel OSS@D Process Design, qui n'offre pas de fichiers d'aide contextuelle.

Declercq, T. (2000) *OSSAD et UML : représentation des concepts OSSAD dans le langage de modélisation unifié*. Document interne IDHEAP.

Favre, E. (2000) *Méthodes pour la conception de systèmes d'information : OSSAD vs (UML, RUP)*. Travail de semestre du cours Management de l'information, IDHEAP.