



HAL
open science

Modélisation et instanciation de processus sur des solutions techniquement hétérogènes

Thierry Vercruyse

► **To cite this version:**

Thierry Vercruyse. Modélisation et instanciation de processus sur des solutions techniquement hétérogènes. Ingénierie, finance et science [cs.CE]. 2015. dumas-01803731

HAL Id: dumas-01803731

<https://dumas.ccsd.cnrs.fr/dumas-01803731>

Submitted on 30 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

CONSERVATOIRE NATIONAL DES ARTS ET METIERS

Toulouse

MEMOIRE

présenté en vue d'obtenir

le DIPLOME D'INGENIEUR CNAM

SPECIALITE : Informatique

OPTION : Architecture et Intégration des Systèmes

par

Thierry VERCRUYSE

Modélisation et instanciation de processus sur des solutions techniquement
hétérogènes

Soutenu le 02 Juillet 2015

JURY

PRESIDENT : Mme. Anne WEI

MEMBRES : Mr. Thierry MILLAN Responsable Pédagogique
Mr. Xavier CREGUT Tuteur
Mr. Emmanuel HAARDT Directeur Associé X2i
Mr. Yannick STOLF Directeur des Opérations X2i

Remerciements

Je remercie,

Florence Sicre, ma compagne et bien-aimée, pour son inébranlable soutien et sa patience, pour m'avoir accompagné pendant la rédaction du mémoire et encouragé tout au long de mon parcours au CNAM. Egalement, mes deux petites filles aux yeux pétillants et curieux qui m'ont porté par leur fierté et leur amour inconditionnel.

Xavier Crégut, mon tuteur, qui a réussi à organiser et structurer mes idées avec un engagement fort dans la préparation du mémoire et de sa soutenance.

Thierry Millan, responsable informatique au CNAM, pour avoir démontré qu'il est possible de faire l'impossible par son implication dans l'accomplissement du mémoire dans son ensemble.

Yannick Stolf, directeur des opérations chez X2i, pour m'avoir donné le temps, le support nécessaire pour accomplir ce mémoire et de m'avoir fait confiance pour mener à bien ces projets. Ainsi que Emmanuel Haardt, directeur associé de X2i, pour son recul et les mises en perspectives éclairantes qui émergent de nos discussions, me permettant de prendre de la hauteur.

Madame Anne WEI, présidente du jury, pour la lecture du mémoire et son investissement dans cette étape finale du cursus d'ingénieur du CNAM.

Liste des abréviations

Abréviations	Signification
BPM	B usiness P rocess M anagement
BPMN	B usiness P rocess M odel and N otation ¹
GED	G estion E lectronique de D ocument
B2B	B usiness to B usiness
UML	U nified M odeling L anguage
OMG	O bject M anagement G roup
IHM	I nterface H omme M achine
NS	EEA6 B usiness P rocess N umerical S tudy
P1	Sous-processus P1 , ou phase 1 du processus NS
BPMN DI	BPMN D ialog I nterchange (DI) Meta-model
XMI	X ML M etadata I nterchange
TDD	T est D riven D evelopment
API	A pplication P rogramming I nterface
REST	R epresentational S tate T ransfer

Traduction des termes anglais

Terme ou expression en anglais	Traduction française
Business Process Management	Gestion des processus métiers.
Business Process Model and Notation	Notation et modèles de processus d'entreprise.
Workflow	Automatisation des processus.
Unified Modeling Language	Langage de modélisation unifié.
Flow Elements (BPMN2.0)	Ensemble d'éléments décrivant la sémantique d'exécution d'une représentation graphique d'un processus en BPMN 2.0.
Activities (BPMN2.0)	Élément graphique BPMN 2.0 représentant une activité d'un processus.
Events (BPMN2.0)	Élément graphique BPMN 2.0 représentant un événement d'un processus.
Gateway (BPMN2.0)	Élément graphique BPMN 2.0 représentant une passerelle permettant de modéliser les décisions, fusions, fourches et jointures d'un processus.
Sequence Flow (BPMN2.0)	Élément graphique BPMN 2.0 représentant l'ordre dans lesquels les activités sont exécutées dans un processus.
Lane (BPMN2.0)	Élément graphique BPMN 2.0 représentant l'organisation des éléments graphiques par rôle représenté par un couloir horizontal.

¹ La signification d'origine « Business Process Modeling Notation » a changée à partir de la version 2.x du standard BPMN et c'est celle-ci qui est retenue maintenant.

User Task (BPMN 2.0)	Tâche (ou activité) nécessitant une intervention humaine par le biais d'une IHM.
Framework	En informatique c'est une structure logicielle ou un ensemble cohérent de composants logiciels qui sert de fondation.
« Peak » de charge	Pic de montée en charge d'une infrastructure informatique. Dans le contexte du commerce électronique, un exemple concret serait l'ouverture des soldes.
Business Process Management Lifecycle	Cycle BPM d'amélioration continue d'un processus entre les étapes de gouvernance d'un processus et la responsabilité d'un processus.
Model	En modélisation, représente un modèle de données.
Boundary	En modélisation, représente une frontière de type IHM ou service.
Controller	En modélisation, représente l'exécution d'une action ou d'un comportement.
Namespace (XML)	Un espace de noms XML est une recommandation du W3C qui permet d'employer des éléments et des attributs nommés dans une instance XML (1)
XML Metadata Interchange	Format d'échanges de métadonnées d'objet de modélisation UML, maintenue par l'OMG, s'appuyant sur le standard XML.
Application Programming Interface	Interface de programmation informatique permettant d'exposer des services à des logiciels tiers.
API REST	Architecture logicielle permettant d'exposer des services logiciels sur des réseaux.

Glossaire

BPMI : ou Business Process Management Initiative, est un consortium de normalisation des processus d'entreprise.

OMG : ou Object Management Group est une association américaine à but non lucratif dont l'objectif est de standardiser et de promouvoir le modèle objet sous toutes ses formes. L'OMG est à la base du standard de modélisation UML. La branche BPM du consortium BPMI a fusionné avec l'OMG en 2005.

Norme ISO 9000:2000 : Norme issue de l'organisation internationale de normalisation qui décrit les principes essentiels des systèmes de management et de qualité. Cette norme a été révisée par l'ISO 9000:2005.

BPMN : Notation graphique standardisée afin de pouvoir modéliser des processus d'entreprise, initialement développée par le BPMI puis maintenant maintenue par l'OMG. La version actuelle est la 2.0.2 depuis décembre 2013.

BPMN 2.0 : Version 2 de la notation BPMN, orientée humains et machines.

XML-BPMN 2.0 : Langage XML sous-jacent d'une notation BPMN 2.0. Ce langage est extensible.

Moteur de Workflow : Composant logiciel capable d'instancier et d'exécuter une définition de processus de manière automatique.

Scrum : Une méthode de gestion de projet informatique agile qui s'appuie sur le découpage d'un projet en « boîtes de temps » ou *sprints*. Les méthodes agiles sont basées sur le principe qu'il vaut mieux adapter que prédire.

Boîte noire : En informatique c'est la représentation d'un système sans considérer son fonctionnement interne. Le fonctionnement d'une boîte noire n'est appréhendé que par ses interactions et associations avec d'autres éléments d'un système. En génie des processus nous pouvons simplifier la représentation d'un processus sous la forme d'une boîte noire sans en décrire sa procédure.

Procédure : Selon la définition AFNOR, contrairement à un processus qui est un ensemble d'activités, une procédure est la manière spécifiée d'accomplir une activité.

Variable d'exécution (processus) : des variables locales accessibles que pendant l'exécution d'une activité.

Variable de processus (processus) : des variables accessibles globalement pendant toute l'exécution d'un processus dans un moteur de workflow.

Graphe de flot de contrôle : une représentation sous forme de graphe de tous les chemins pouvant être parcourus par un programme informatique pendant son exécution. Il permet de s'assurer d'une bonne couverture des tests.

Table des matières

Remerciements	2
Liste des abréviations	3
Traduction des termes anglais	3
Glossaire	5
Table des matières	7
I INTRODUCTION	9
II QU'EST-CE QU'UN PROCESSUS METIER EN ENTREPRISE	11
II.1 DEFINITION	11
II.1.1 Exemple d'un processus simple	12
II.1.2 Types de processus métiers en entreprise	13
II.2 PROBLEMATIQUE D'HOMOGENEISATION	13
II.3 PROBLEMATIQUE D'HETEROGENEITE TECHNOLOGIQUE	14
III MODELISATION DE PROCESSUS.....	15
III.1 DESCRIPTION DE PROCESSUS AVEC BPMN 2.0.....	16
III.1.1 Eléments syntaxiques	17
III.1.2 Définition de processus	18
III.2 LES OUTILS POUR DEFINIR LES PROCESSUS	20
III.2.1 Activiti Modeler.....	20
III.2.2 Enterprise Architect	22
III.3 METHODOLOGIE	23
III.3.1 Cycle BPM	24
III.3.2 Méthodologie ICONIX	27
III.3.3 Modèles de processus.....	32
IV INSTANCIATION DE PROCESSUS.....	44
IV.1 ANATOMIE DE LA REPRESENTATION XML-BPMN 2.0	45
IV.1.1 La définition de processus	45
IV.1.2 La représentation graphique de processus.....	46
IV.1.3 Interopérabilité des processus basé sur le standard XML	46
IV.2 CAS 1 : AIRBUS ACOUSTIC WORKFLOWS.....	48
IV.2.1 Processus d'études numériques	48
IV.2.2 Les dysfonctionnements	51
IV.2.3 Objectif de la solution proposée	51
IV.2.4 Organisation du projet.....	52
IV.3 LA PLATE-FORME ACTIVITI	55
IV.3.1 Activiti Engine	57
IV.3.2 Activiti Designer	57
IV.3.3 Activiti Explorer	58
IV.4 METHODOLOGIE	60
IV.4.1 Intégration dans Activiti Engine	61
IV.4.2 Exécution dans Alfresco.....	69
IV.5 BILAN.....	77
V INSTANCIATION DANS UN CONTEXTE HETEROGENE	81
V.1 CAS 2 : DESCOURS & CABAUD ET LES CIRCUITS DE VALIDATION	82
V.2 APPLICATION DE LA METHODOLOGIE.....	83

V.3	INTEGRATION ET EXECUTION.....	86
V.3.1	Rétro-ingénierie et moteur de workflow sur mesures.....	86
V.3.2	Développement piloté par les tests	89
V.3.3	Interfaces homme machine	90
V.3.4	Suivi du processus	91
V.3.5	Optimisations	91
V.4	BILAN.....	92
VI	CONCLUSION.....	94
	Annexe 1 : Etapes de la méthodologie appliquée – Synthèse	97
	Annexe 2 : Exemple des deux niveaux de BPMN 2.0 (graphique et XML exécutable).....	98
	Annexe 3 : Cartographie initiale du processus NS-P1	99
	Annexe 4 : Formalisation en BPMN 2.0 du processus NS-P1 dans Activiti	100
	Annexe 5 : Captures d’écrans du processus NS-P1 dans Alfresco-Share	101
	Annexe 6 : Eléments graphiques du standard BPMN 2.0	102

I Introduction

Suite à mon cursus au CNAM², j'ai évolué au poste de directeur technique dans la société X2i³ située à Toulouse et m'intéresse très fortement à tous les sujets portant sur l'industrialisation de la production de projets informatiques. Deux projets sur lesquels j'ai participé portaient sur l'informatisation de processus humains et j'ai décidé de les utiliser comme support des travaux présentés dans ce mémoire.

Tous les jours, peut être sans s'en rendre compte, nous observons des processus. Prenons l'exemple d'une pile de documents où il faut relire, accepter ou refuser chaque document, possiblement en demandant une révision à un tiers et ainsi de suite pour aboutir à une publication finale. Ou bien le suivi de l'escalade d'une demande de budget faite à un service de financement et l'attente d'une réponse sans savoir si elle a été traitée. Ou encore quand nous achetons un livre précis sur internet, imaginons toutes les étapes qui se succèdent entre le règlement, l'expédition et la livraison de l'article puis éventuellement toutes les erreurs qui se sont produites quand nous en recevons un autre par mégarde. Un processus non informatisé est très souvent chronophage et puisqu'il est réalisé par des humains, est potentiellement source d'erreurs. Sans support informatique, nous manquons de visibilité sur son avancement, il peut être répétitif et dépendre de plusieurs services ce qui entraîne de l'inefficacité ainsi que des frustrations.

En entreprise, la production d'un service ou d'un bien est généralement issue de l'enchaînement de différentes tâches. Ces tâches faites par différentes personnes incarnent un processus métier d'une fonction de cette entreprise ou plus simplement un processus métier d'entreprise. Elles ont besoin d'être maîtrisées et sont généralement réalisées avec l'aide de supports logiciels dans le meilleur des cas.

Qu'est-ce qu'un processus ? La norme de qualité *ISO [9000:2000]* (2) le définit ainsi : un processus est un « Ensemble d'activités corrélées ou interactives qui transforme des éléments d'entrée en éléments de sortie ».

En entreprise, un processus métier est une série de tâches que des humains (le personnel de l'entreprise) et/ou des machines (programmes informatiques) doivent réaliser en séquence dans le temps pour créer de la valeur ajoutée pour ses clients et les parties prenantes. Dans ce contexte, les termes processus métier en entreprise et workflows d'entreprise sont souvent utilisés comme synonymes.

Cette définition place l'accent sur les points suivants : un processus peut être segmenté en une séquence de plusieurs activités, ces activités doivent être réalisées ou interprétées par quelqu'un ou quelque chose et l'objectif final est d'apporter une valeur ajoutée à l'entreprise directement ou indirectement.

Quels peuvent être les apports de l'informatisation d'un processus métier ? Je pourrais désigner trois éléments de réponse à cette question : la réduction du coût, l'amélioration de

² Suite à une reconversion professionnelle en 2007, j'ai obtenu une VAE me permettant de m'inscrire au CNAM pour suivre le cursus AISL en 2011.

³ X2i est une société d'ingénierie en informatique située à Toulouse depuis 2005, spécialisée dans le commerce électronique et compte 50 salariés.

la qualité du service et la réduction des durées de traitements. En conséquence, les erreurs sont moins fréquentes et cela dégage du temps à l'initiateur du processus pour faire un travail à plus forte valeur ajoutée. L'automatisation est une partie intrinsèque de l'industrialisation et l'informatisation est un moyen d'y parvenir.

Comment informatiser un processus métier ? Il existe un bon nombre de méthodes suivant différentes normes plus ou moins lourdes et facilement applicables. Je vais vous présenter dans ce mémoire une méthode de modélisation et d'intégration de processus métier d'entreprise par le biais d'outils informatiques dans le cadre d'un projet puis son application sur un autre projet technologiquement hétérogène. Ce mémoire indiquera pourquoi cette problématique d'hétérogénéité est importante et pourquoi il faut tenter de s'orienter vers des solutions unifiées en identifiant leurs limites.

Je vous présenterais un premier cas d'étude ayant pour objectif la démonstration de la faisabilité de l'informatisation d'un processus humain d'études numériques avioniques à but d'industrialisation et la méthodologie qui a été appliquée pour y parvenir. Avec comme point de départ l'analyse de ce processus par la récolte des informations lors des discussions et échanges avec les membres de l'équipe. A l'issue de ce travail, une solution technique est proposée puis implémentée dans le logiciel de gestion de contenus de l'équipe.

Un deuxième cas d'étude additionnel sera ensuite présenté qui abordera l'application de cette méthodologie dans un contexte technologiquement très différent, celui du commerce électronique visant la validation de demandes de transactions « professionnel à professionnel » ou « B2B ».

Ce mémoire est structuré de la manière suivante.

- La première partie présente une définition de processus métier en entreprise ainsi que les problématiques abordées dans ce mémoire.
- La deuxième partie montre un moyen de modéliser les processus métier en s'appuyant sur le standard de notation BPMN 2.0.
- La troisième partie indique comment un processus peut être instancié sur une solution informatique.
- La quatrième partie expose l'instanciation de processus dans un contexte technologique différent.
- Une conclusion viendra présenter un bilan.

II Qu'est-ce qu'un processus métier en entreprise

Le terme processus métier est présent depuis le début du 20^{ème} siècle dans un effort de rationalisation de processus de fabrication appelé ingénierie des processus.

Commençons par quelques concepts de base concernant les processus métiers en entreprise.

II.1 Définition

En complément de la définition faite dans l'introduction, il est essentiel de savoir comment identifier un processus métier en entreprise et quels sont les critères qui le décrivent.

Un processus métier en entreprise est identifiable principalement par les critères suivants.

- Il a un objectif, un but.
- Il a des entrants spécifiques d'information.
- Il est instancié par des événements.
- Il a des sorties spécifiques, il produit quelque chose.
- Il utilise des ressources (machine et/ou humain).
- Il a un certain nombre d'activités qui sont réalisées par des échanges dans un certain ordre.
- Il peut affecter plusieurs unités organisationnelles et de manière horizontale.
- Il crée de la valeur ajoutée pour le client. Le client peut être externe ou interne.

Pourquoi est-ce important d'adopter un formalisme ? Il existe plusieurs formalismes pour représenter un processus métier en entreprise avec ces critères.

En 2000 un modèle de représentation de processus métier a été proposé par Hans-Erik Eriksson et Magnus Penker (3). Ce modèle représente les activités d'un processus sous la forme de *boîte noire*, il permet de simplifier la représentation d'un processus en indiquant l'événement qui le déclenche, les associations entrantes, sortantes, les objectifs et les ressources impliquées sans en décrire le fonctionnement interne. Le format de ce modèle permet de présenter un processus dans son ensemble à très haut-niveau, et son concept est une extension de la modélisation UML. Le diagramme de la Figure 1 ci-dessous présente un modèle générique de processus selon le concept d'*Eriksson-Penker*.

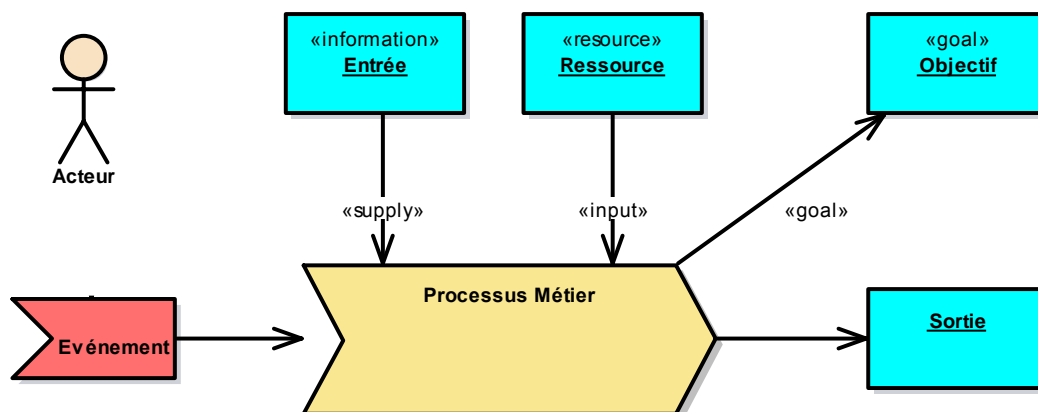


Figure 1- Représentation d'un processus métier selon le concept d'Eriksson-Penker.

II.1.1 Exemple d'un processus simple

Afin de mieux appréhender cette syntaxe de formalisation d'un processus, vous pouvez trouver ci-après, avec la Figure 2, une représentation avec la notation *Eriksson-Penker* d'un exemple de processus familial suivant.

« Je retire un colis à la poste : A la réception d'un avis de colis, le destinataire fournit un avis de réception et une pièce d'identité valide à un employé du bureau de poste. Celui-ci lui transmet le colis et le colis est retiré »

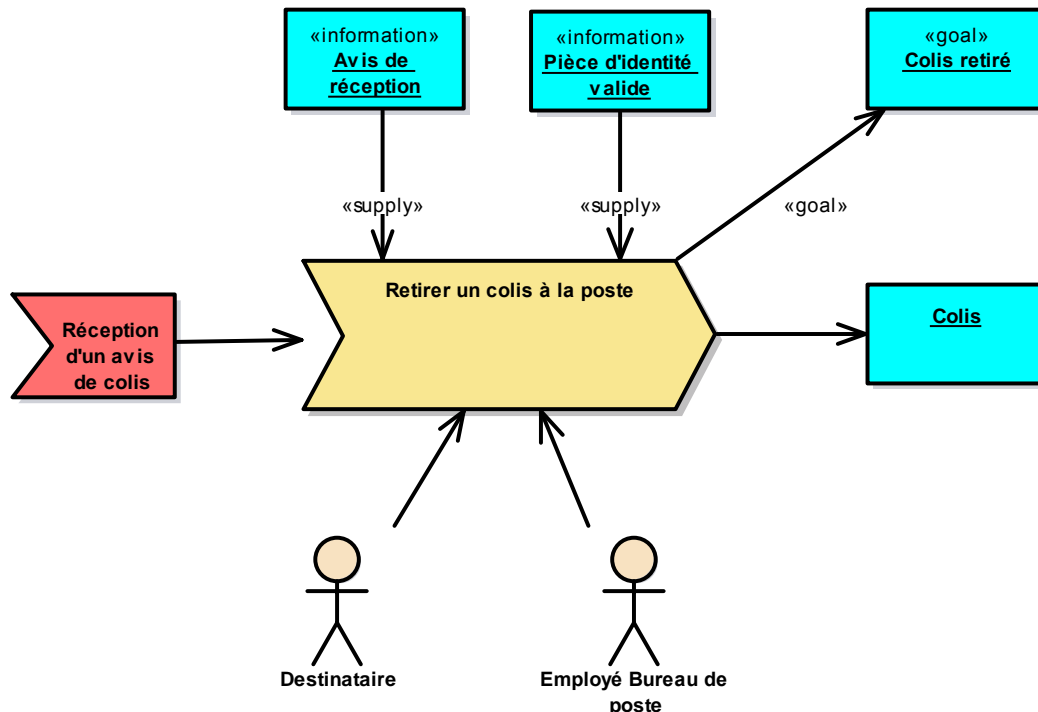


Figure 2 - Exemple d'un processus simple en notation Eriksson-Penker.

Cette représentation d'un processus utilisant le méta-modèle de la Figure 1, permet de rapidement observer quel est l'événement déclencheur du processus « Réception d'un avis de colis » que le processus à deux entrants « Avis de réception » et « Pièce d'identité » les ressources humaines identifiées sont le « Destinataire » et « l'Employé du Bureau de poste » et qu'il a pour objectif le « retrait du colis » puis finalement il produit un « colis ». Cependant il est incomplet, plusieurs questions se posent. On ne sait pas qui est responsable de quoi ? Le produit est destiné à qui ? L'objectif a une valeur pour qui ? Quels sont les différentes activités de ce processus ?

La représentation *Eriksson-Penker* atteint vite ses limites (manque de précisions sur les détails du processus, manque de détails sur les responsabilités et les relations des ressources impliquées) mais elle est un bon point de départ pour la transformation d'un processus initialement sous la forme d'une idée, d'un besoin, en modèles de processus précis. La représentation *Eriksson-Penker* est une vue très élevée et synthétique d'un processus permettant d'initier les bonnes interrogations et s'intègre bien dans la cartographie macroscopique de la vue métier de processus d'entreprise.

II.1.2 Types de processus métiers en entreprise

Il existe deux types de processus métier d'entreprise, les **processus opérationnels** et les **processus de soutiens**. Les deux sont encadrées par une **gestion de processus**.

Les processus opérationnels constituent les activités métier de l'entreprise et apporte de la valeur ajoutée. Les processus de soutiens supportent les processus opérationnels et la gestion des processus représentent la gouvernance opérationnelle d'une entreprise.

II.2 Problématique d'homogénéisation

Nous pouvons représenter un processus chacun à sa manière, néanmoins afin de faciliter la modélisation d'un processus l'objectif est d'adopter un formalisme clair avec une notation simple adaptée aux processus et qui sera acceptée par les différents intervenants. Il est difficile de contrôler ce qui n'est pas compris.

Pourquoi est-il important d'adopter un modèle de formalisation pendant la modélisation de processus et quel modèle est adapté pour l'exécution des processus ?

Imaginons une situation de réunion de travail où chaque personne dans une équipe communique avec un langage différent (anglais, espagnol, etc.) à l'issue ils quittent certainement la réunion avec des mauvaises interprétations de ce que chaque orateur avait l'intention d'exprimer. Cette problématique est aussi présente dans la modélisation de processus d'entreprise. Les résultats peuvent être catastrophiques si pendant la définition de processus, chaque personne interprète le besoin et la conception différemment.

Dès lors que nous adoptons un modèle de formalisme, il devient aisé de communiquer et de collaborer autour d'un processus pour ensuite l'ajuster et l'optimiser avec un vocabulaire commun. C'est une désambiguïsation qui consiste à supprimer les ambiguïtés contenues dans la communication de l'information.

Un des critères de choix sur le formalisme est le non rejet de certains clients, nous avons besoin d'une expression simple et précise de la dynamique des processus. En grande partie c'est pour cette raison que la notation BPMN 2.0 existe et a été appliquée sur les projets de ce mémoire puisqu'il est adapté pour la modélisation et l'instanciation de processus.

Un des axes d'amélioration pour aboutir à un logiciel de qualité est l'adoption d'un modèle de formalisme. C'est un langage avec un vocabulaire commun, formalisé, précis, qui est connu et adopté par tous les membres de l'équipe et il est donc, à ce titre, un vecteur favorisé pour communiquer sans ambiguïté. A qui s'adresse ce modèle de formalisme ?

Les différents points de vue à qui ce formalisme s'adresse sont en partie les suivants.

- Le point de vue **métier** : les stratégies, ressources et objectifs.
- Le point de vue **fonctionnel** : décomposition fonctionnelle du processus et des interactions.
- Le point de vue **technologique** : définir les technologies logicielles utilisées.

Cette communication est essentielle pour aboutir à une compréhension commune de la problématique traitée par les différentes intervenants (le client, le responsable métier jusqu'à l'équipe de production du logiciel).

En plus de l'adoption d'un modèle formalisé, dans le cadre d'une société de services en informatique, il est essentiel de pouvoir capitaliser à la fois sur les méthodes et sur les moyens d'intégration informatique dans des solutions techniquement hétérogènes.

II.3 Problématique d'hétérogénéité technologique

Le terme hétérogène est utilisé ici pour définir des technologies ayant une nature différentes. Un projet informatique web en Java à une nature différente qu'un projet informatique web en PHP, les concepts et protocoles de communication sont similaires, certes (besoin de composants traitant une requête sur le protocole http produisant une réponse sous la forme d'une page web sur laquelle il est possible d'interagir et d'une couche de persistance de données), mais l'implémentation est techniquement distincte. Cette nature différente est de l'ordre des compétences techniques requises et des composants d'architecture applicative et infrastructurelles mises en œuvre pour l'implémentation logicielle.

Comment instancier un modèle de processus sur des environnements logiciels techniquement hétérogènes ? Quels sont les moyens pour aboutir à une interopérabilité entre la définition de processus métier formalisée et son instanciation logicielle.

L'écosystème informatique est très vaste et comporte une multitude d'outils pour instancier et suivre des processus humains. Chaque entreprise choisit son outil en fonction de critères de sélection qui lui sont propres : coût, support, documentation, formations, faculté d'adoption, intégration dans le système d'information existant, etc.

Il en devient impossible d'instancier (par implémentation) un processus de manière générique sur toutes les solutions logicielles, mais il est possible de construire un modèle de processus unifié puis de la projeter différemment sur les solutions logicielles.

Le défi est de réduire cette divergence de projection technologique au minimum, c'est l'interopérabilité. La solution Activiti présentée dans ce mémoire offre la possibilité de s'appuyer sur un protocole de communication commun concernant l'exécution de processus.

Un premier cas d'implémentation de processus présentera un projet ayant des contraintes technologiques au niveau du logiciel de gestion de contenu d'entreprise Alfresco⁴ utilisé par ses équipes. Puis un deuxième cas d'implémentation de processus sera présenté sur une plate-forme de commerce électronique techniquement hétérogène et dépourvue initialement de solution d'exécution de processus.

Les deux projets ont utilisé la même méthode de formalisation de processus mais l'implémentation et l'exécution a été différente.

⁴ Alfresco est un logiciel de gestion de contenu d'entreprise créée par *Alfresco Software* en 2005 sous licence libre, basé sur la technologie Java.

III Modélisation de processus

La modélisation de processus est une étape de la **transposition⁵ d'un processus métier** aboutissant en son instanciation et exécution. Cette transposition est une étape importante qui permet la portabilité d'un processus sur des technologies hétérogènes.

Un modèle de processus métier issu d'une idée ou d'un besoin est transposé au format BPMN 2.0 puis la définition de processus qui en résulte est adaptée pour qu'il puisse être exécutable. La Figure 3 présente d'une part les étapes de cette transposition et d'autre part (cadre rouge) ou se situe cette partie du mémoire dans cet enchainement d'étapes.

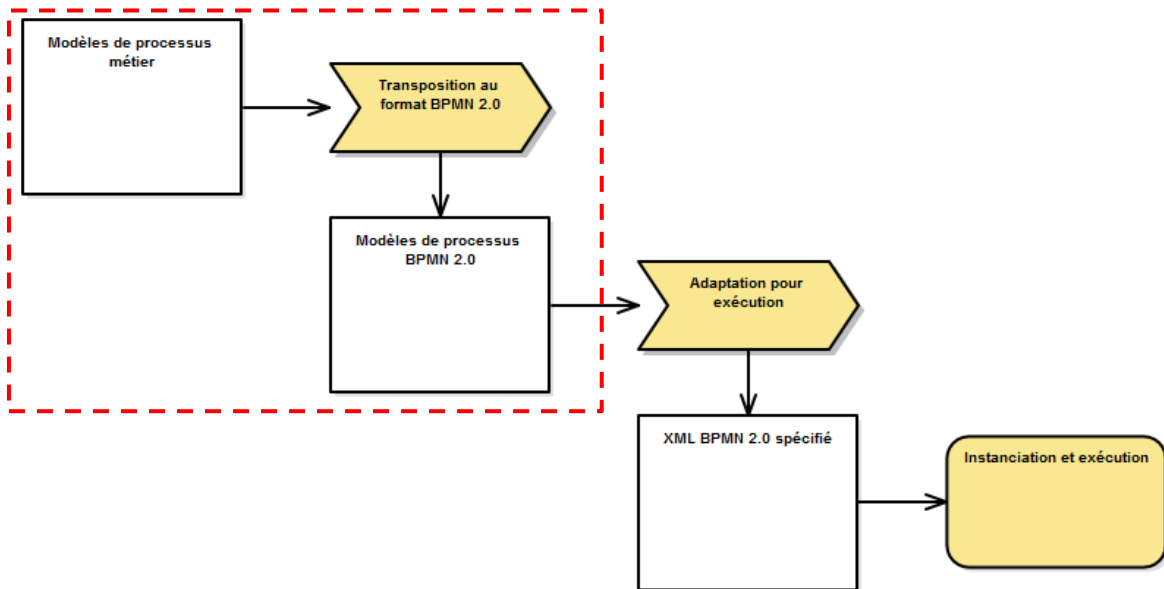


Figure 3 - Etapes de transposition d'un processus métier - focus modélisation.

La transposition d'un processus permet de le spécifier et cette spécification est un gage d'efficacité et permet une facilitation de l'apprentissage. Elle devient un référentiel et sert de base pour toutes communications de processus. Ce référentiel est structuré et documenté et il faut être certain que la méthode de modélisation soit correcte dans le sens où elle est applicable, adoptée et permet d'obtenir des résultats exploitables. Ensuite il faut s'assurer que la méthode ait été appliquée dans les modèles et que le processus soit compris.

Afin de pouvoir transposer un processus au format BPMN 2.0 je vais commencer par exposer comment ce formalisme décrit les processus et quels sont les apports de cette notation.

La réussite de la modélisation de processus métiers dépend essentiellement des choix méthodologies et techniques. Cette partie terminera par une méthodologie qui a été mise en œuvre intégrant la collaboration des parties prenantes durant le cycle de vie du projet.

⁵ Transposition dans le sens où la description de processus est adaptée pour respecter la syntaxe du formalisme BPMN 2.0.

III.1 Description de processus avec BPMN 2.0



La notation BPMN 2.0 est une spécification de notation de processus métiers d'entreprise maintenue par l'OMG et initialement développée par le BPPI. Cette notation est une syntaxe formalisée et standardisée pour décrire des processus métiers. La version actuelle est la 2.0.2 depuis décembre 2013⁶.

Cette spécification vise à la fois :

- Les **humains** : par sa notation graphique de processus métier ;
- Les **machines** : par sa syntaxe sous-jacente, en langage de balisage générique XML utilisée par des interfaces de logiciels, décrivant le processus métier. Cette syntaxe décrit à la fois le processus et sa représentation graphique. Mais pas que, elle est extensible et est un moyen d'aboutir à une interopérabilité de l'instanciation de processus.

L'objectif principal de BPMN est de fournir une notation qui est compréhensible par chaque intervenant du projet informatique : de l'analyste métier qui rédige la version initiale du processus, au client qui validera le processus, au développeur technique qui est responsable de l'implémentation technologique du logiciel qui exécutera ces processus puis finalement aux managers qui géreront et contrôleront l'efficacité du processus.

Un autre objectif est que le langage XML conçu pour l'exécution des processus métiers peut être visualisé avec une notation commune et intégré avec une interface de programmation dans les logiciels. Seule l'implémentation de cette interface peut varier selon les technologies. Nous allons voir que pour le projet *Acoustic*⁷, nous avons fait le choix d'encapsuler dans un service le moteur informatique d'exécution du processus ce qui a permis de réduire le périmètre d'intégration spécifique et d'augmenter le périmètre d'intégration réutilisable.

La spécification BPMN 2.0 est divisée en plusieurs types de diagrammes (Processus, Collaboration, Orchestration, et Conversation) (4) qui sont utilisés chacun pour exprimer un aspect précis de processus métiers, ce mémoire va se concentrer uniquement sur les diagrammes de **Processus** BPMN 2.0.

En résumé, la notation BPMN 2.0 permet d'un part une **modélisation** graphique basée sur un standard de notation (4) et d'autre part une **implémentation** technologique basée sur une approche d'interopérabilité par le biais de la norme de standard ouvert XML.

La Figure 4 indique un rapide historique de l'évolution du standard BPMN (5).

⁶ Ce mémoire ne rentrera pas dans les détails de la spécification BPMN 2.0 mais vous pourriez la consulter librement sur le site de l'OMG : <http://www.omg.org/spec/BPMN/2.0.2/>.

⁷ Cf. chapitre IV.2 « Cas 1 : Airbus Acoustic Workflows ».

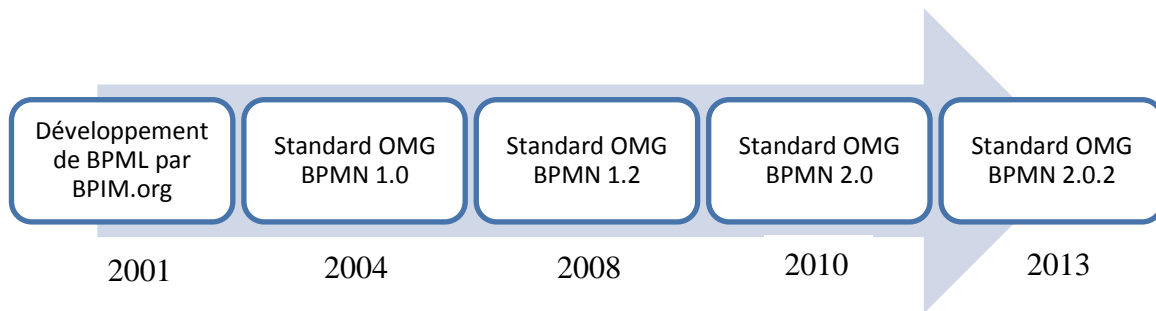


Figure 4 - Historique de l'évolution du standard BPMN.

Il existe à ce jour deux versions de BPMN : la 1 et la 2. La version 1 est plus orientée métier tandis que la version 2 est en plus orientée technique. A partir de la version 2, BPMN apporte d'une part une amélioration de la version 1 et d'autre part des éléments de modélisation informatiques facilitant son exécution dans des moteurs de workflow.

L'abréviation BPMN à elle aussi évolué entre la version 1.x et 2.x :

<i>Version</i>	<i>Signification</i>
1.x	Business Process Modeling Notation
2.x	Business Process Model and Notation

III.1.1 Eléments syntaxiques

Quels sont les éléments de syntaxe de cette norme pour en faire une représentation graphique ?

En BPMN 2 un processus est décrit comme un graphe qui est une série de 4 catégories d'éléments de bases définissant chacun une sémantiques d'exécution fini.





	Activities : représentation d'une activité dans un processus
	Gateways : représentation d'un branchement (ou passerelle) permettant de modéliser les décisions, fusions, fourches et jointures d'un processus
	Events : représentation d'un événement dans un processus
	Sequence Flows : représentation de l'ordre dans lesquels les activités sont exécutées dans un processus

Figure 5 - Eléments de base BPMN 2.0.

Ces éléments de bases peuvent être enrichis de **symboles iconographiques** pour les spécialiser apportant une flexibilité dans la modélisation. Par exemple pour typer les activités ou ajouter un marqueur de boucle ou de sous-processus. Vous pouvez trouver en

annexe ⁸ de ce document un poster représentant tous les éléments graphiques de la syntaxe BPMN 2.0. Ils sont nommés *Flow Elements* et incarnent les concepts de base de ce langage.

Reprenons l'exemple de la Figure 2 de la section II.1.1 du chapitre précédant formalisé avec la notation BPMN 2.0.

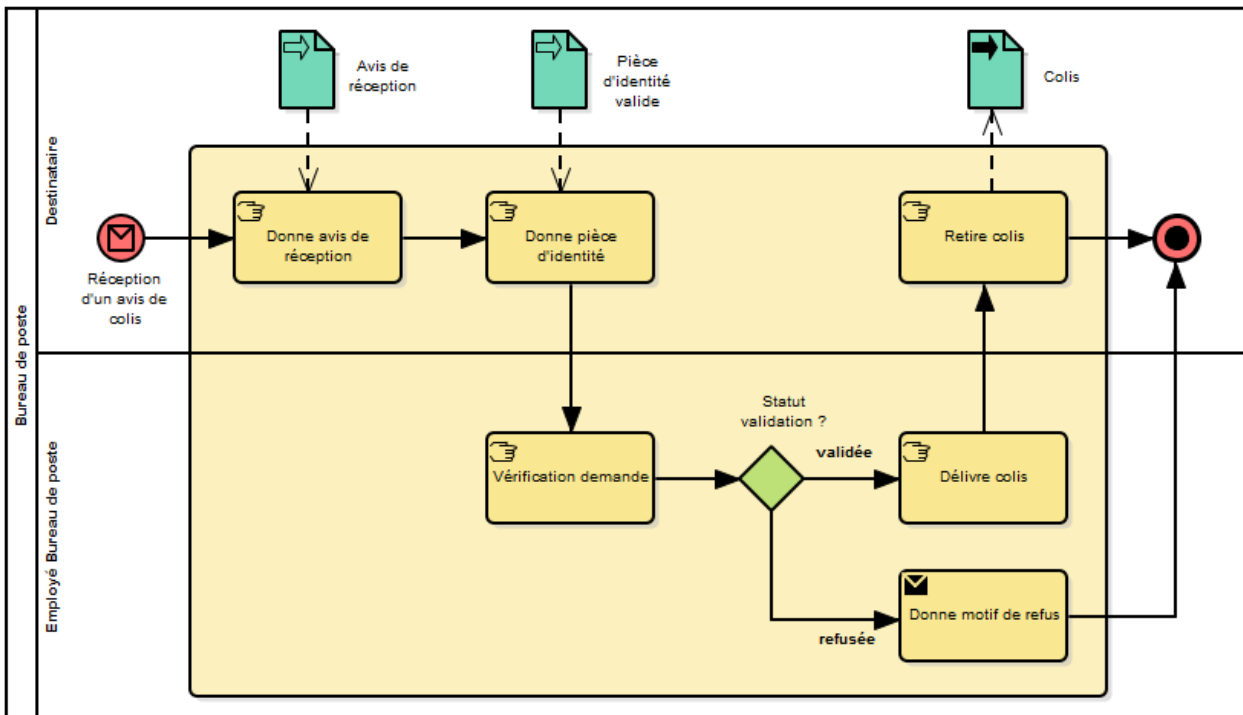


Figure 6 - Exemple d'un processus simple en notation BPMN 2.0.

Ici nous apercevons les détails du processus, jusqu'à présent représentés sous la forme de *boîte noire*, avec l'enchaînement des activités et la passerelle de choix exclusive « statut validation ». Puis l'ensemble des activités faites par les rôles est précisé en utilisant des couloirs horizontaux appelés *Lanes*. On comprend bien que l'événement déclencheur et le « colis » sont destinés au destinataire.

III.1.2 Définition de processus

Dès lors qu'un processus est formalisé graphiquement il sera possible de le rendre exécutable par instantiation de la **définition de processus**. L'OMG a prévu que la spécification BPMN 2.0 puisse être souple et ils ont ajouté la possibilité d'étendre cette spécification par des points d'extensions. Ces points d'extensions permettent d'associer des données spécifiques aux éléments graphiques directement dans le XML qui sont ensuite exploitables par les moteurs d'exécution de processus (ou moteur de workflow). Ce n'est pas la définition de processus qui est exécutée mais c'est le moteur de workflow qui instancie un nouveau processus à partir de cette définition.

Le choix principal sur l'utilisation de BPMN 2.0 sur le projet *Acoustic* était la réduction de l'écart entre la spécification d'un processus et l'exécution informatique de celui-ci. Il nous

⁸ Annexe 3 : Eléments graphiques du standard BPMN 2.0

a été possible de formaliser un processus puis d'utiliser le XML sous-jacent pour l'exécuter avec le moteur de workflow Activiti. Ce moteur exploite les points d'extensions prévus dans la spécification BPMN 2.0 notamment pour ajouter des définitions de formulaires pour la création d'IHM et de variables servants pour l'exécution.

Les bénéfices de l'utilisation d'un moteur de workflow compatible BPMN 2.0 sont les suivants :

1. Quand les responsables métiers collaborent sur des projets de processus métier ils peuvent utiliser un langage commun pour communiquer sur le processus sans prendre en compte la solution qui sera finalement utilisée pour implémenter son exécution.
2. N'importe quel outil sachant produire de l'XML compatible BPMN 2.0 peut être utilisé et théoriquement n'importe quel moteur de workflow compatible BPMN 2.0 pourra aussi être utilisé pour exécuter le processus.

L'utilisation d'un moteur de workflow compatible BPMN 2.0 permet de modéliser et d'exécuter un processus sur des solutions technologiquement hétérogènes. La restriction ici est la compatibilité de la solution avec le moteur de workflow.

La spécification BPMN 2.0 a trois niveaux superposés de détails, chaque niveau apporte un certain nombre de fonctionnalités pouvant être modélisées et leur représentation graphique. Le niveau **basique** (ou simple) n'est utile que pour des besoins de prototypage rapide par exemple une cartographie initiale. Le niveau **cœur** (ou descriptif) couvre 80% de la spécification BPMN 2.0 et est suffisante pour modéliser et exécuter la majorité des cas. Le niveau **avancé** (DoDAF⁹ et complète) propose des fonctionnalités qui sont en réalité rarement utilisées.

Une carte heuristique¹⁰ des fonctionnalités du niveau **cœur** est représentée dans la Figure 7 ci-dessous :

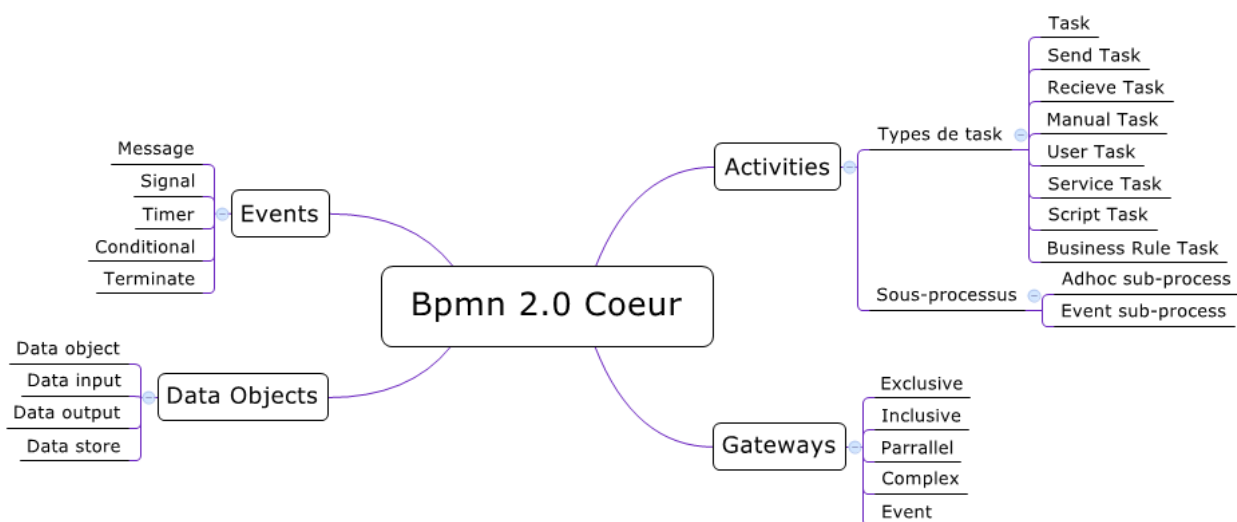


Figure 7 - Carté heuristique du cœur de la spécification BPMN 2.0.

⁹ DoDAF : acronyme pour Department of Defense Architecture Framework

¹⁰ « Mind Map » en anglais : est un schéma permettant de présenter les liens entre un sujet et des concepts ou idées.

Chaque moteur de workflow compatible avec la spécification BPMN 2.0 n'en couvre qu'une certaine partie. Ce qui veut dire que seulement un sous-ensemble de toute la spécification BPMN 2.0 sera supporté par un moteur de workflow donné.

Le choix du moteur de workflow repose sur son niveau de couverture de la spécification BPMN 2.0.

Pendant la phase d'études du projet *Acoustic*, j'ai réussi à démontrer la faisabilité de la modélisation, instanciation et exécution de processus. Puis j'ai présenté les concepts et méthodes de rédaction d'un diagramme de processus en BPMN 2.0 au responsable métier du client habitué à d'autres outils comme *Visio*¹¹. Afin qu'il puisse rapidement monter en compétence sur le formalisme BPMN 2.0 je lui ai fourni plusieurs « schémas type » : boucles de validation, tâches en parallèle, décisions et gestion des erreurs et un tutoriel sur Powerpoint contenant les éléments de base servant de référence pour formaliser un processus en BPMN 2.0 avec Activiti.

Le chapitre suivant vous dévoilera les différents outils utilisés dans la cadre du projet *Acoustic* qui m'ont permis de modéliser des processus.

III.2 Les outils pour définir les processus

Cette section présente les différents outils utilisés afin de pouvoir définir les processus par modélisation.

III.2.1 Activiti Modeler

L'outil Activiti Modeler de la plate-forme Activiti¹² est une application web permettant l'édition de processus sous forme graphique compatible BPMN 2.0. Cet outil permet aux analystes métier de définir les premières versions d'un processus en mode collaboratif.

Dans le cadre de mes activités chez la société X2i, je suis souvent amené à devoir faire de la modélisation UML et nous utilisons un outil commun : *Enterprise Architect*¹³. Mon souhait était de rester sur cet outil dont la modélisation est versionnable et collaborative d'autant plus qu'il permet d'importer et d'exporter des définitions XML-BPMN 2.0. Malheureusement les points d'extensions utilisés par Activiti pour gérer les formulaires d'interactions homme/machine ne sont pas gérés par Enterprise Architect.

Les premières versions du processus du projet *Acoustic* ont été modélisées avec Enterprise Architect par habitude puis importées dans Activiti Modeler, Figure 8, pour continuer sa définition pour le moteur d'exécution Activiti. L'outil Activiti Modeler couvre largement les besoins du responsable métier du projet pour modéliser les définitions simples de processus.

Les besoins couverts par Activiti Modeler pour un responsable métier sont les suivants :

- Capacité à modéliser des processus au format BPMN 2.0 avec un outil centralisé.
- Capacité à collaborer sur la modélisation d'un processus.

¹¹ Logiciel de création de diagrammes ou d'organigrammes édité par *Microsoft*.

¹² La plate-forme Activiti sera présentée intégralement dans la section « IV.3 La plate-forme Activiti ».

¹³ Cf. section III.2.2

- Capacité à renseigner les différentes propriétés de formulaires ainsi que les variables d'exécution de processus d'un point de vue métier.
- Possibilité d'exporter et importer une définition de processus au format XML afin de la compléter dans des outils permettant son exécution.

Le responsable métier a été formé pour être autonome sur cet outil avec comme support le référentiel BPMN 2.0 que j'avais rédigé lors de la phase d'étude.

Par-contre l'outil Activiti Modeler est limité dès qu'il est nécessaire de formaliser l'exécution de processus¹⁴. Cet aspect de la modélisation est de l'ordre technique et l'outil Activiti Designer est mieux adapté.

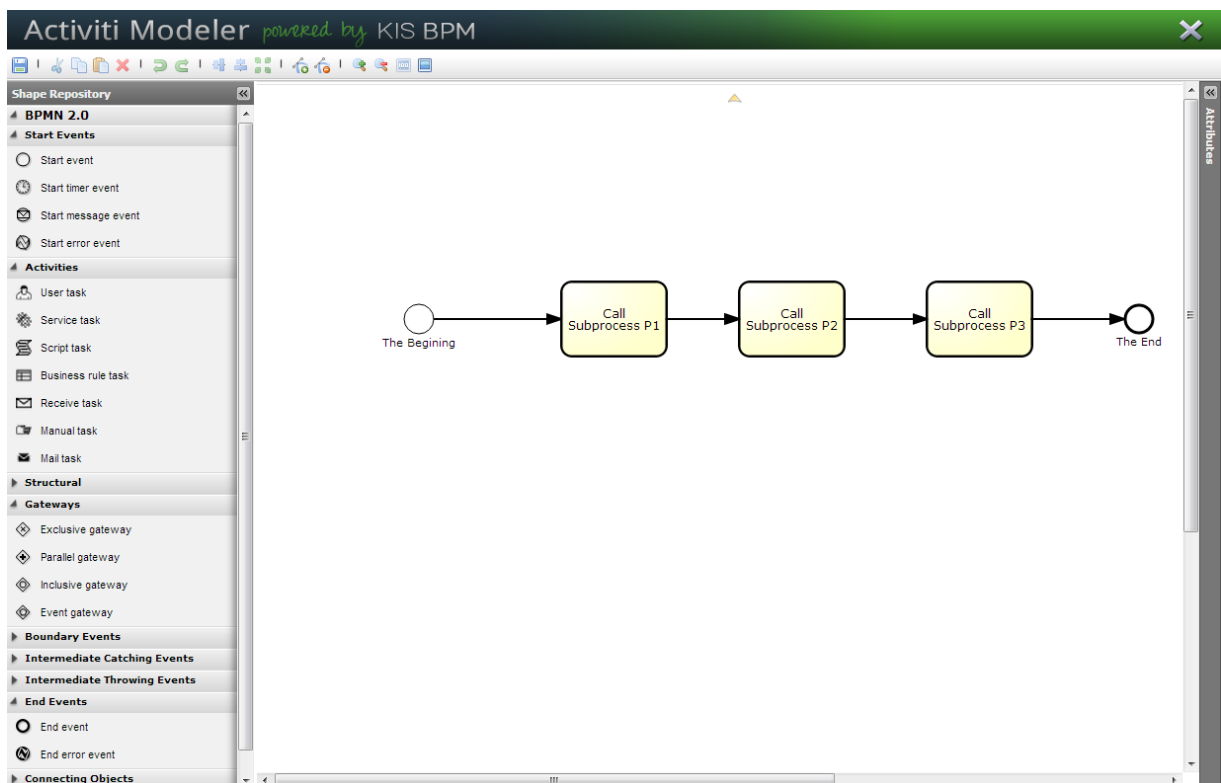


Figure 8 – Ecran de modélisation de processus avec le logiciel web Activiti Modeler.

¹⁴ Cf. la partie IV « Instanciation de processus ».

III.2.2 Enterprise Architect

Enterprise Architect (6) est un logiciel de modélisation et de conception UML, couvrant toutes les étapes du cycle de développement d'applications. Il est édité par la société australienne *Sparx Systems* avec une licence logicielle commerciale. Enterprise Architect intègre tous les diagrammes de la spécification UML ainsi que de nombreuses extensions comme *BPMN 2.0*, *Archimate 2*, *SysML*, *Eriksson-Penker*, *ERD*, etc.

Il est principalement capable de :

- Modéliser des diagrammes UML et BPMN 2.0.
- Simuler une exécution de diagramme d'état ou d'activité.
- Valider le formalisme de diagrammes BPMN 2.0.
- Générer du code source dans de nombreuses technologies (Java, C, C#, C++, PHP, Python, Delphi, ActionScript, ArcGIS et VisualBasic).
- Réaliser des transformations MDA.
- Publier de la documentation dans différents formats (Word, PDF et HTML).
- Générer des diagrammes de classes à partir de code source.
- Naviguer dans les relations des éléments de modélisation.
- Tracer des modèles avec des exigences.
- Travailler en équipe en mode collaboratif sur un entrepôt commun.
- Maintenir un historique avec un système de contrôle des versions qui permet de fusionner différentes révisions d'un modèle.

La Figure 9 présente l'espace de travail de l'outil Enterprise Architect.

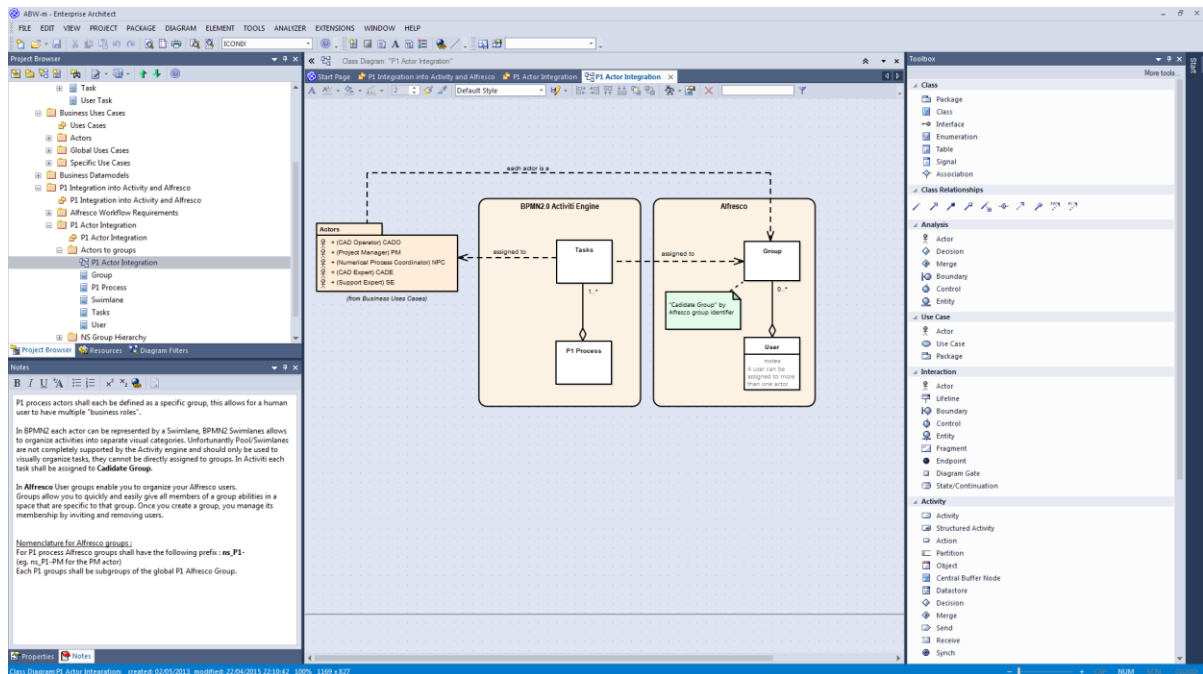


Figure 9 – Ecran de modélisation UML avec le logiciel Enterprise Architect 12.

Cet outil est très bien documenté et est doté d'une communauté active d'architectes qui aident à l'améliorer constamment (7). Ayant testé de nombreux outils de modélisation abordables, je n'ai encore rien découvert d'aussi complet, même si son utilisation est parfois compliquée.

III.3 Méthodologie

Un projet informatique, quelle que soit son ampleur et ses objectifs, nécessite la mise en place d'un planning organisationnel tout au long de son cycle de vie. C'est ainsi qu'est apparue la notion de méthode dans la discipline de gestion de projet par exemple.

Dans le contexte informatique des processus, une méthode, peut être définie comme une « démarche fournissant une méthodologie et des notations standards qui aident à concevoir des processus de qualité ».

En génie logiciel il existe plusieurs méthodes d'analyse et de conception, les plus populaires sont les méthodes **objet** et **agiles**. Toutes nécessitent un effort différent dans les axes de documentation, d'apprentissage, de lourdeur, de rigueur et de coût. Par-contre toutes partagent le même objectif : livrer un logiciel de qualité qui satisfait le client (où le maître d'ouvrage) selon les facteurs définissant la qualité du génie logiciel. Je pourrais citer les facteurs suivants.

- **Validité** : aptitude au logiciel produit à répondre aux fonctions définies dans le document de spécifications du client.
- **Extensibilité** : facilité avec laquelle le logiciel évolue et est modifié dans les cycles de maintenance du logiciel.
- **Fiabilité** : capacité au logiciel de fonctionner dans des conditions anormales cadrées par le document de spécifications du client.
- **Vérifiabilité** : facilité avec laquelle le logiciel est testable selon les procédures de test du projet informatique dans lequel il est produit.
- **Efficacité** : utilisation optimale des ressources matériels (mémoire disponible, cycles processeur, fiabilité face aux « peaks » de charges, etc.).
- **Compatibilité** : facilité avec laquelle un logiciel peut être intégré avec d'autres logiciels.
- **Réutilisabilité** : capitalisation en tout en en partie du logiciel pour être réutilisé dans des applications différentes.
- **Portabilité** : facilité avec laquelle le logiciel pourra être intégré dans des environnements matériels et logiciels hétérogènes.
- **Intégrité** : aptitude d'un logiciel à protéger son code et ses données contre des accès non autorisés. Notamment dans le contexte de sécurisation des logiciels sur internet.
- **Facilité d'emploi** : facilité d'apprentissage, d'utilisation, de préparation des données, d'interprétation des erreurs et de rattrapage en cas d'erreur d'utilisation.
- **Retour sur investissement (ROI)** : aptitude au logiciel produit à respecter l'enveloppe budgétaire fixée par le maître d'œuvre et le maître d'ouvrage.

Le choix de la pondération de chaque facteur dépend du contexte. Le choix de la méthode de génie logiciel dépend à la fois du contexte et des facteurs de qualité. Selon mon expérience, toute méthodologie de génie logiciel doit prendre en considération une pondération de ces facteurs en fonction d'un contexte.

Ce chapitre du mémoire va vous décrire la méthodologie de modélisation qui a été appliquée sur le projet *Acoustic* puis sur le projet *Descours & Cabaud* et comment les différents étapes ont été traitées. La méthodologie que je vous présente est pilotée par le cycle d'amélioration continu BPM et utilise la méthodologie ICONIX comme guide pour la modélisation et l'instanciation. Elle a pour objectif d'être légère, facile d'emploi et réutilisable.

III.3.1 Cycle BPM

En génie logiciel, une production de logiciel s'appuie sur une méthodologie et suit un cycle de vie organisationnel. Ce cycle de vie décrit toutes les étapes du développement du logiciel de sa conception à sa disparition. Dans le contexte des processus métier d'entreprise, un processus n'est pas toujours informatisé mais doit suivre un cycle d'amélioration continu.

Le BPM (8), ou la gestion des processus métier, est une discipline dont l'objectif est d'aboutir à une vue globale améliorée des processus métiers d'une entreprise et de leur interactions afin de les optimiser. En entreprise, tous les processus ne peuvent pas être informatisés et en conséquence le cycle BPM doit pouvoir s'adapter à tous types de processus qu'ils soient implémentés sur des solutions logiciels ou pas.

Le cycle BPM, ou *Business Process Management Lifecycle*, est un cycle d'amélioration continu en 5 étapes permettant d'optimiser un processus. Comme toute démarche d'amélioration continue, l'objectif est d'avoir une méthode structurée et de mettre en œuvre les solutions les plus adaptées et pérennes. Je pourrais l'introduire comme une projection de la roue de *Deming*, utilisée dans les normes de qualité ISO, apposée sur les principes de la gestion de processus métiers.

La création d'un processus métier inclut l'enchaînement cyclique de 5 étapes, chaque étape représente une phase importante de la construction de la solution :

1. Conception
2. Modélisation
3. Exécution
4. Suivi
5. Optimisation

Le cycle BPM est un composant central de la gestion métier des processus. Il permet de gérer chaque phase de la mise en œuvre d'un processus. BPM n'est pas une suite de logiciels permettant d'implémenter des processus c'est une discipline avec une approche systémique. Nous distinguons dans le cycle BPM deux niveaux : la **gouvernance de processus** qui doit rechercher et optimiser l'intégration des processus d'une entreprise et la **responsabilité d'un processus** qui doit mettre en place et contrôler le processus avec des ressources compétentes. Le cycle BPM est un cycle d'amélioration continu et dans ce sens à chaque changement du processus, le cycle recommence.

Les diagrammes de la page suivante¹⁵ montrent la correspondance entre le cycle de *Deming* et le cycle BPM.

¹⁵ La première étape de chaque cycle est soulignée.

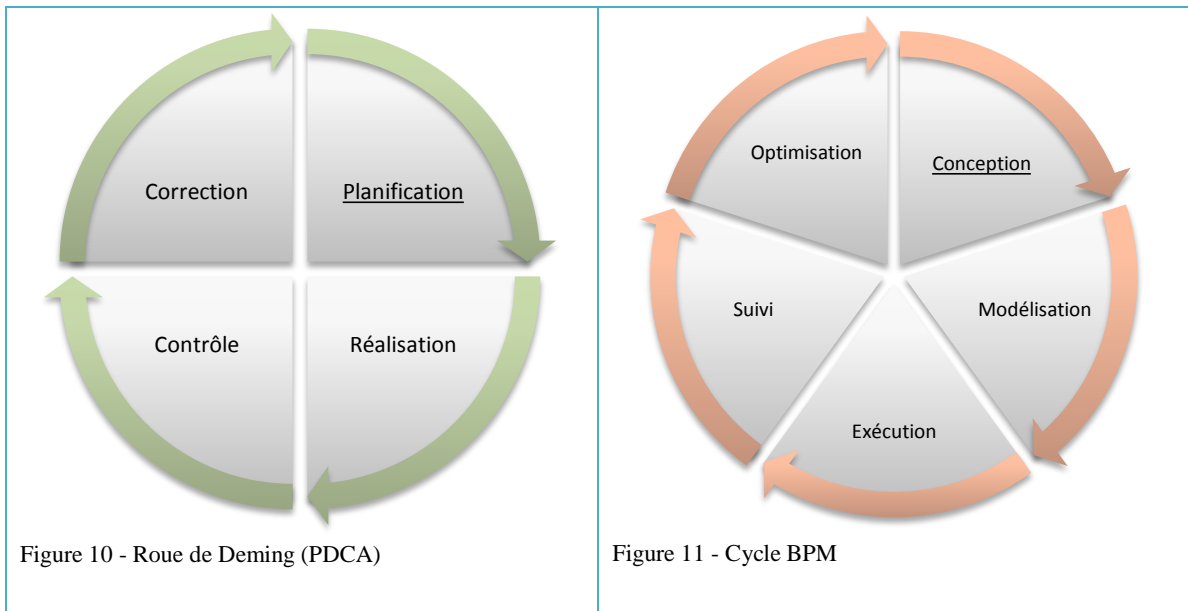


Figure 10 - Roue de Deming (PDCA)

Figure 11 - Cycle BPM

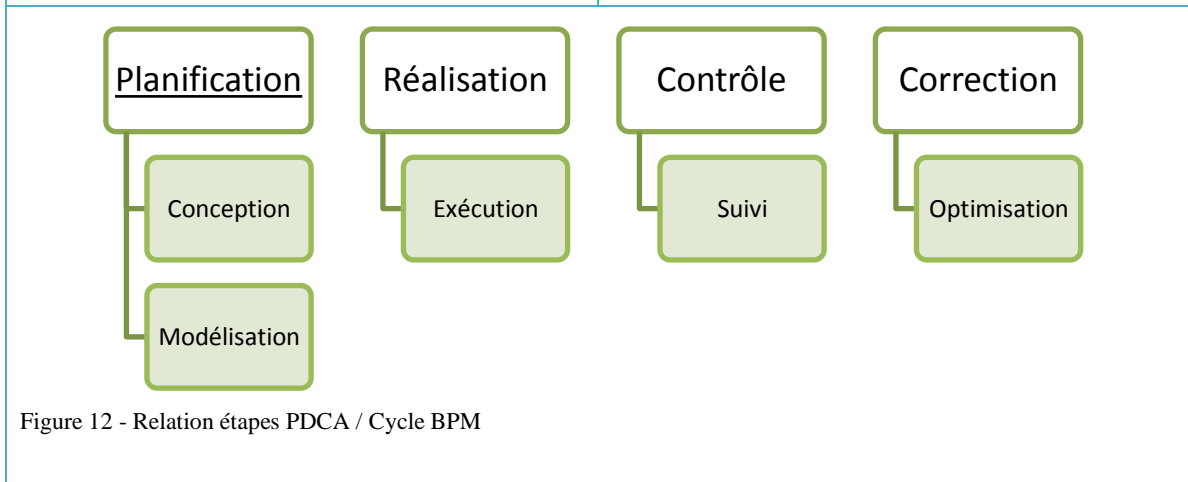


Figure 12 - Relation étapes PDCA / Cycle BPM

Nous allons explorer chaque étape du cycle BPM dans les sous-sections suivantes.

III.3.1.1 Conception

L'étape de conception inventorie toutes les activités qui ont pour résultat la spécification du processus. Dans cette étape nous allons pouvoir : identifier les activités de haut-niveau, discuter autour de potentiels changements organisationnels, définir les contrats de niveau de service¹⁶ et spécifier des détails de processus tel que les acteurs, notifications et en termes BPMN 2.0, les escalades de responsabilité vers d'autres acteurs.

Cette étape est très souvent révélatrice des dysfonctionnements et incohérences d'un processus existant ou du niveau de visibilité pour un processus nouveau. Simplement il suffit de poser le problème pour en avoir une vision plus claire. A mon sens c'est une étape fondamentale qui requière plusieurs itérations avec le client.

Elle permet de poser un langage commun et les bases des premiers modèles de processus. Dans cette étape il n'est pas nécessaire d'imposer un formalisme BPMN 2.0 strict.

¹⁶ SLA (ou Service Level Agreement) en anglais.

L'essentiel est de produire rapidement une modélisation du processus sans rentrer dans les détails.

Dans cette étape nous devons trouver les réponses aux questions suivantes, bien entendu adaptables en fonction du contexte client.

- Qui fait quoi ? Pourquoi ?
- Quel est le but du processus ?
- Quel est le rôle de chaque personne intervenant sur le processus ?
- Comment se font les échanges d'informations ?
- Quels sont vos problèmes, nœuds d'étranglements actuels ?
- Comment revenir en arrière ?
- Quels sont les documents qui seront véhiculés au cours de ce processus ?
- Quels sont les entrants et les sortants ?
- ...

Tout en formalisant une synthèse des éléments identifiés qui serviront de référence aux prochaines étapes. La conception demande à appréhender un processus dans son ensemble sans pour autant se concentrer sur les détails techniques de son exécution.

III.3.1.2 Modélisation

Le processus métier est spécifié et validé.

L'étape de modélisation permet d'appliquer le formalisme strict BPMN 2.0 au processus. Le processus est précisé dans ses détails. Les activités sont précisées et on différencie les activités humaines des activités machines (script, envoi de messages ou services). Les activités humaines sont typées et on précise lesquelles sont manuelles et lesquelles nécessitent une IHM. Nous identifions la logique d'enchaînement des activités : celles qui peuvent être faites en parallèle, si pour commencer une activité nous avons besoin que certaines se terminent, etc. Nous identifions les cas d'erreur.

Cette étape a besoin d'être faite avec le support d'un expert BPMN 2.0.

Dans cette étape nous devons trouver les réponses aux questions suivantes.

- Que se passe-t-il en cas d'erreur ?
- Qu'est-ce qui se passe si ... ?
- Existe-il des interfaces de saisie ?
- Existe-t-il des similitudes dans le processus qui peuvent être factorisées en sous-processus ?
- Quels sont les indicateurs de performances ? Comment les mesurer ?

L'objectif est de simuler le processus et de penser aux cas anormaux puis de l'adapter.

Je n'ai pas réussi à ce jour à trouver un outil avec lequel une simulation peut être faite, l'outil de modélisation UML, Enterprise Architect, est doté d'une fonctionnalité de simulation pour les diagrammes d'état et d'activités, que j'utilise, mais ne fonctionne pas pour le BPMN 2.0. Activiti Explorer est trop spécifique à ce stade de l'informatisation du processus pour être retenue dans la méthode décrite.

C'est la dernière étape avant la première frontière d'adaptation du processus BPMN 2.0. Suite à la modélisation d'un processus un choix doit être fait celui de l'exécuter

manuellement ou d'utiliser une solution logicielle qui s'appuie sur un moteur d'exécution de workflow.

III.3.1.3 Exécution

Cette étape exécute le processus. Des instances de processus sont lancées et elles sont utilisées par les utilisateurs finaux. Des solutions logicielles peuvent être mises en pratique pour exécuter le processus avec une combinaison possible d'intervention humaine ce qui demande un effort sur la documentation du processus.

Cette étape doit permettre de suivre l'avancement des activités d'une instance d'un processus.

III.3.1.4 Suivi

Cette étape mesure les indicateurs de performances définit et la performance du processus.

Des rapports sont générés puis analysés et comparés avec les contrats de services.

Dans cette étape nous devons trouver les réponses aux questions suivantes.

- Quels sont les retours des utilisateurs ?
- Existe-t-il des goulots d'étranglement ?
- Existe-t-il des axes d'amélioration ?
- Le processus répond-t-il au besoin ?

III.3.1.5 Optimisation

Cette dernière étape permet de réduire les incohérences et inefficiences identifiées pendant l'étape de suivi puis d'améliorer le processus. L'objectif de cette étape est d'éventuellement réduire le coût du processus. Globalement un effort est fait pour optimiser la valeur ajoutée du processus.

III.3.2 Méthodologie ICONIX

ICONIX (9) (10) est une méthodologie de génie logiciel qui se base uniquement sur un sous ensemble de quatre diagrammes UML (Cas d'utilisations, Classes, Robustesse et Séquences). Elle se base sur le concept suivant : «En théorie chaque aspect de la modélisation UML est utile, mais en pratique, seulement un sous-ensemble est réellement utilisé par manque de temps et de moyens.». *Doug Rosenberg* a mis au point cette méthodologie suite à de nombreuses observations faites au cours de sa carrière. Il a fondé en 1984 la société *ICONIX Software Engineering*, ayant comme mission la formation et le support de réalisation de projet informatique pour des sociétés dans les secteurs d'activités suivants : aérospatiale, militaire, semi-conducteurs, énergie, services tiers et fabrication. Personnellement il a réalisé des projets informatiques pour les sociétés suivantes : *Hughes Radar Systems Group, Hughes Research Labs, TRW Microelectronics Center, CALMA Corporation, and The University of Southern California Electrical Engineering*

Department. Cette méthodologie mélange des notions essentielles des méthodologies RUP¹⁷ et agile.

Durant mon expérience professionnelle en tant qu'architecte logiciel pour la plate-forme de commerce électronique *Magento* chez X2i, j'ai recherché une méthodologie de modélisation informatique efficiente. Dans le cadre de projets informatique de type commerce électronique, en général, le rapport entre le budget du projet et celui de la conception est de l'ordre de 10% et encore souvent la conception est considérée comme étant inutile. La technologie *Magento* est très spécifique et les développeurs expérimentés sont rares. Ma problématique était de pouvoir séparer le domaine du problème (le quoi) du domaine de la solution (le comment) et de pouvoir générer la structure des couches logicielles dans laquelle le projet *Magento* allait se réaliser tout en ayant une méthodologie légère. Cela permet de garantir que technologiquement l'architecture *Magento* est respectée.

III.3.2.1 Domaine

Cette méthodologie suit un enchaînement d'étapes itératives et agiles. Commençant par la définition d'un vocabulaire commun : **le Domaine**. Le domaine est modélisé avec un diagramme de classes UML mais limitant les types d'associations afin de se concentrer sur l'essentiel : la définition des termes, leur définition et l'association entre les termes. Chaque terme est modélisé par une Classe. Les deux relations autorisées entre les classes sont listées ci-après.

- **Agrégation** (*losange clair*) : relation entre termes décrivant une appartenance utilisant la syntaxe : un terme « avoir un » un autre terme. Par exemple « Le processus P1 a un *Numacous Repository* qui a un *Business Request* ».
- **Héritage** (*triangle*) : relation entre termes décrivant une généalogie utilisant la syntaxe : un terme « est un » autre terme. Par exemple « Une carte bancaire est un moyen de paiement ».

Ce glossaire, constitué de noms, est ensuite maintenu et utilisé comme vocabulaire commun pour toute la suite de la conception. Cela permet de supprimer toute ambiguïté. La modélisation du premier diagramme de Domaine doit se faire en moins d'une heure puis il doit être validé par le client qui est placé au cœur du processus de la conception. Le domaine est tiré à partir de la relecture et la validation des spécifications et des premières versions d'IHM.

Par exemple pour le projet *Acoustic*, j'ai décidé dans le cadre du cycle BPM de conception, avant tout, de modéliser le domaine avec le client pendant une réunion de travail avec l'outil de modélisation UML Enterprise Architect. Dans le cadre de ce projet cette phase a permis à chaque membre de l'équipe client de se mettre d'accord sur les termes du projet, ces termes ont été utilisés jusqu'à l'exécution du processus. La Figure 13 montre le domaine¹⁸ du processus phase 1 (ou P1) du projet *Acoustic*.

¹⁷ *Rational Unified Process*, instanciation par *Rational Software* (IBM) des préceptes UP. UP (ou *Unified Process*) est une méthode de développement pour les logiciels orientés objets.

¹⁸ Toutes les documentations et modèles UML du projet *Acoustic* ont été rédigées en anglais.

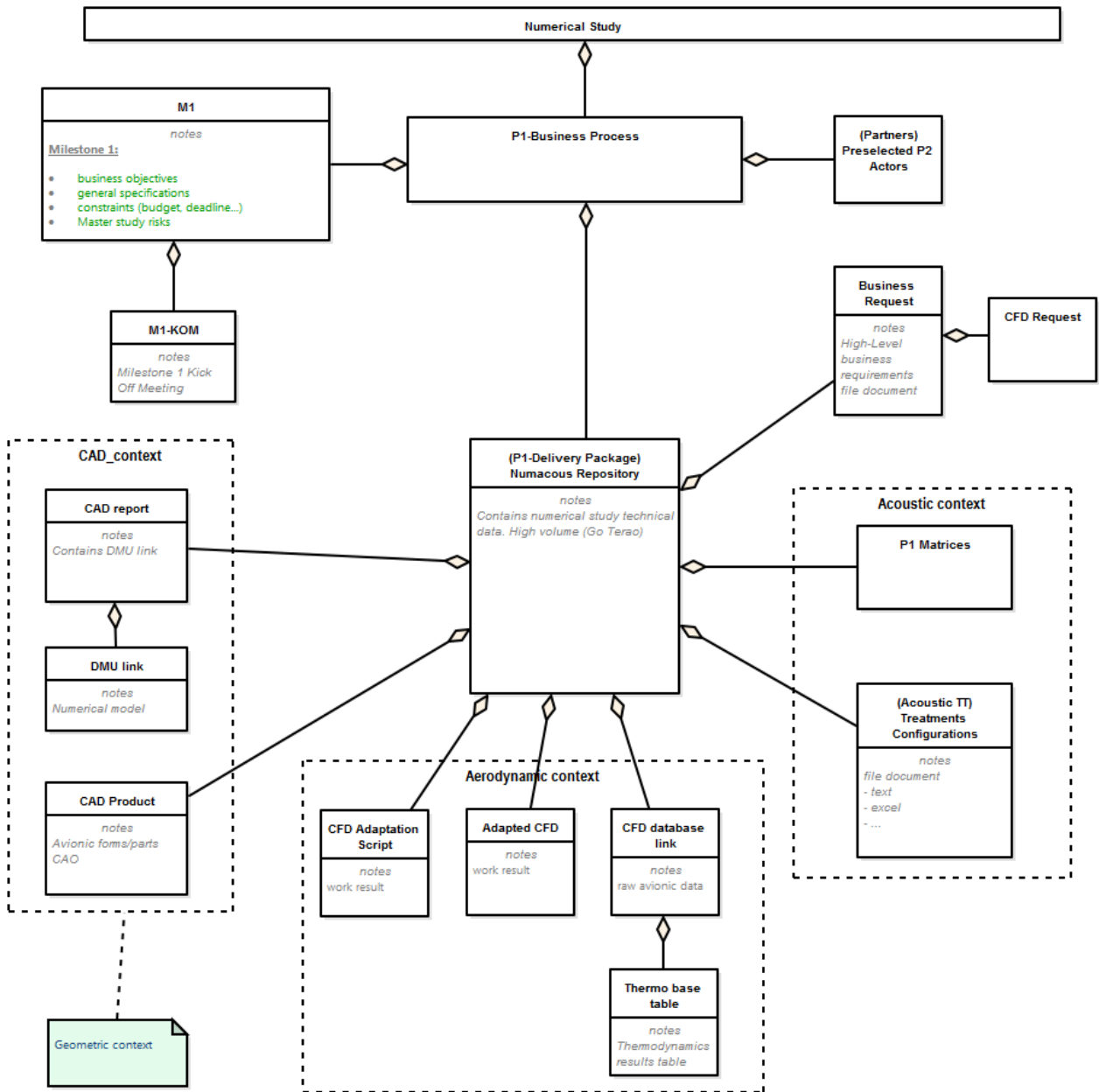


Figure 13 - Diagramme de domaine du processus NS-P1.

III.3.2.2 Cas d'utilisations

Dès que le premier domaine est posé il faut définir les **cas d'utilisations**, ils sont construits en utilisant rigoureusement le vocabulaire du domaine. Chaque cas d'utilisation décrit un scénario nominal et les scénarios exceptionnels (ou conditions anormales). Les cas d'utilisation sont ensuite positionnés dans une organisation logique sous la forme de packages avec l'outil de modélisation. Contrairement au formalisme UML où un diagramme de cas d'utilisation est « statique », en ICONIX une notion de dynamisme est ajouté par l'utilisation de 2 stéréotypes de relations entre les use cases.

- **Invokes** : Au cours du scénario du cas d'utilisation actuel celui associé par le stéréotype *Invokes* sera appelé, sans préciser quand.
- **Precedes** : Le scénario du cas d'utilisation associé par le stéréotype *Precedes* sera appelé avant l'actuel.

Le diagramme de la Figure 14 ci-dessous montre un diagramme de cas d'utilisations du processus P1 du projet *Acoustic*.

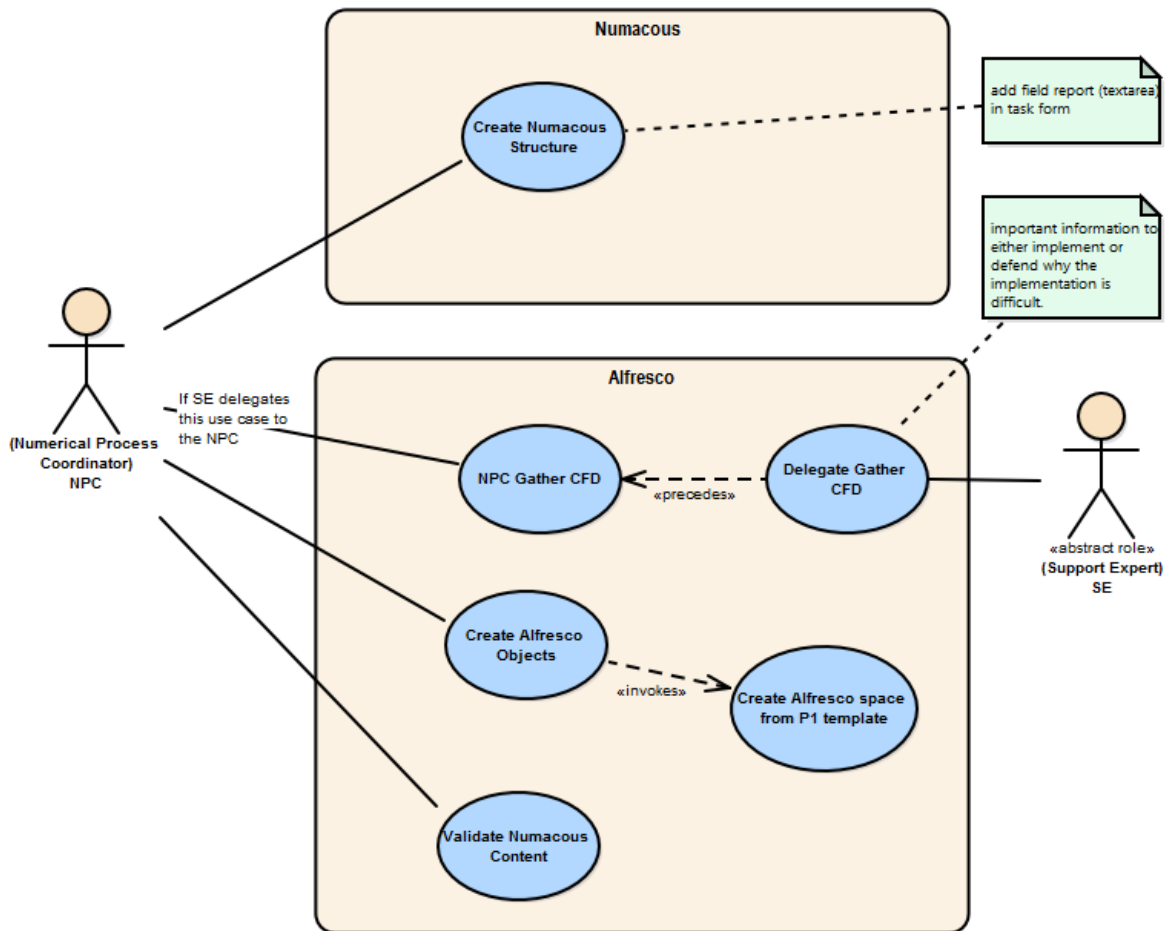


Figure 14 - Processus NS-P1 cas d'utilisation de l'acteur NPC.

A ce stade, très fréquemment le domaine est ajusté parce que l'analyste découvre d'autres termes et à besoin de préciser des termes déjà identifiés.

III.3.2.3 Robustness

Ensuite pour chaque cas d'utilisation un diagramme de *robustness*¹⁹ est réalisé. Il permet de raffiner les cas d'utilisation et de formaliser la syntaxe des scénarios.

Le diagramme de *robustness* utilise 3 types d'éléments pour décrire un fonctionnement *Boundary-Controller-Entity* du domaine du problème (le quoi ?).

- **Entity** : le modèle décrivant une information, par exemple, un client ou une commande. On utilise directement les éléments du domaine.
- **Boundary** : une IHM ou un service tiers.
- **Controller** : l'action qui est réalisée, c'est grossièrement la « colle » entre les *Models* et les *Boundaries*.

¹⁹ Le diagramme de *robustness* est un diagramme d'analyse dynamique proche du diagramme de collaboration UML. Ce diagramme est issu de la technique d'analyse conçu par Ivar Jacobson (un des contributeurs de UML) (18).

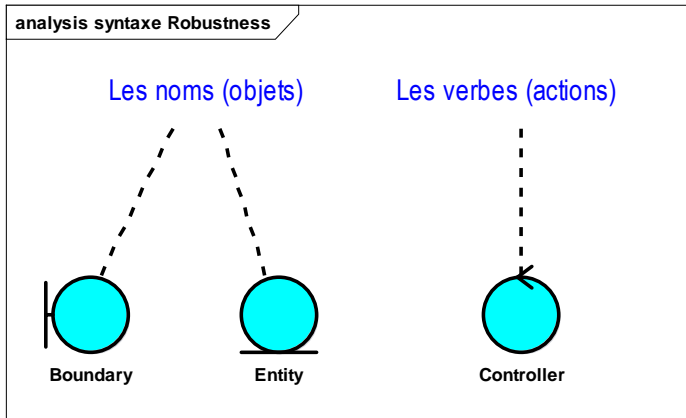


Figure 15 - Symboles de diagramme de *robustness* ICONIX.

La syntaxe des diagrammes de *robustness* doit respecter les règles suivantes.

- Une phrase commence par un nom.
- Un nom peut discuter avec un verbe et *vice versa*.
- Un verbe peut discuter avec un autre verbe.
- Un nom ne peut pas discuter avec un autre nom.

Cela permet de raffiner et préciser les scénarios décrits dans les cas d'utilisations en se basant sur cette syntaxe. Les scénarios rédigés dans les cas d'utilisations permettent ensuite de définir comment l'application sera testée d'un point de vue fonctionnelle.

Les diagrammes de *robustness* sont jetables leur seul objectif est de formaliser et de raffiner les cas d'utilisations et de découvrir des nouveaux cas alternatif.

III.3.2.4 Résumé

La Figure 16 est un extrait du livre « *Use Case Driven Object Modeling With UML (9)* » et résume les notions présentées.

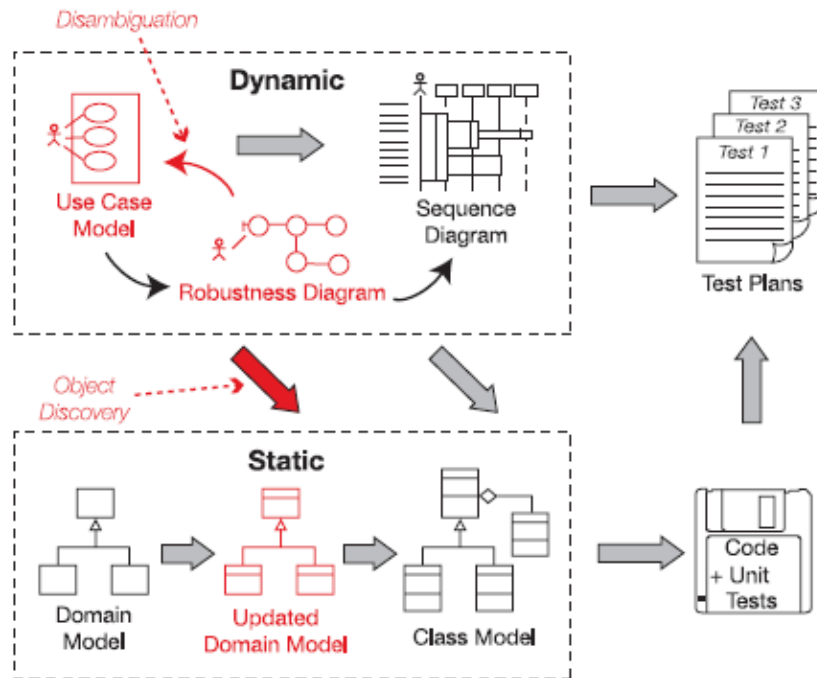


Figure 16 - Synthèse de la méthodologie *ICONIX*

Cette méthodologie a une vraie utilité si elle est appliquée avec la connaissance de la solution logicielle qui sera utilisée. Cela nous permet d'optimiser l'effort de conception.

Dans le cadre de la conception de processus métier je n'ai eu besoin de définir de diagrammes de séquences et d'aller à la conception des modèles de classes pouvant être générés et utilisés pour la création des tests unitaires.

III.3.3 Modèles de processus

La modélisation de l'activité d'une entreprise doit être faite dans un premier temps de manière globale indépendamment d'un moteur de workflow puisque cette étape est faite par des responsables métier qui ne sont pas techniques et s'adresse au client pour validation. Le workflow se concentre sur l'exécution d'une série de tâches démarrées manuellement ou automatiquement et de l'interaction entre elles.

III.3.3.1 Cartographie initiale d'un processus

Ici une cartographie²⁰ est une représentation graphique ou un diagramme voire un croquis des liens entre les différents objets d'un processus à l'étude sans la rigueur d'un formalisme. L'objectif de cette première modélisation est d'obtenir une vision globale, macroscopique et synthétique de ces liens. Elle est réalisée pendant la phase d'étude et souvent par le client ou un non expert en processus. Elle est informelle.

²⁰ Le terme cartographie est à ne pas confondre avec le terme « urbanisation ».

Par exemple pour le projet *Acoustic* le responsable métier du processus NS nous avait fourni un croquis réalisé avec l'outil Visio représentant sa vision du processus existant avec ses termes.

Un autre exemple sur le projet *Descours & Cabaud*, en réunion avec le directeur général de *Debonix*, le chef de projet informatique et le directeur digital e-commerce du groupe *Descours & Cabaud* nous avons dessiné sur un tableau la cartographie du processus issue de l'interview. Puis nous avons reçu un nouvel exemplaire sous forme de diagramme Powerpoint suite à une prise de recul du client.

Ces cartographies sont importantes et servent de base pour la modélisation et elles aident à comprendre comment un processus fonctionne. Par-contre étant donné qu'une cartographie est rédigée par différentes personnes ayant différentes cultures et techniques de modélisation elle est faite sans formalisme ce qui rend sa lecture incertaine. Essentiellement c'est pour cette raison que l'adoption d'un modèle de processus est obligatoire. Du fait que les termes varient entre les différents interlocuteurs, par exemple le mot panier en e-commerce est assimilé à cadis, « cart » ou « checkout » selon les personnes, il est nécessaire d'adopter le vocabulaire commun du domaine afin d'exclure toutes ambiguïtés.

III.3.3.2 Vocabulaire commun et scénarios

Nous commençons par la définition du **domaine** qui est formalisé sous la forme d'un diagramme de classe. Ce diagramme contiendra tous les termes et leurs relations. Chaque terme aura une brève définition. Le domaine est revu et adapté afin de toujours être le reflet de l'état du projet. Plusieurs diagrammes complémentaires peuvent être faits en se focalisant sur des aspects précis du projet afin de simplifier leur lecture, sachant que le diagramme globale peut être très grand, contenant de nombreux termes, et rapidement illisible.

Les termes du domaine seront utilisés rigoureusement dans tous les autres modèles. Des glossaires peuvent être générés sous forme de documentation projet. Ils sont listés, les doublons supprimés et les relations « est-un » « a un » sont faites, à chaque terme une courte définition de quelques lignes est rédigée.

Des intervenants humains peuvent participer à l'exécution d'un processus, pour les identifier, on s'appuie sur les diagrammes de cas d'utilisation UML. Ces diagrammes permettent de répertorier les **acteurs** et leur **rôle** dans le processus. Un diagramme est fait représentant les différents acteurs et leur relation s'il y en a²¹. Un rôle dans un processus est modélisé avec un acteur UML. Nous définissons « qui fait quoi et quels sont leur responsabilités ». Un rôle dans un processus est distinct de la personne réelle qui l'interprète, il peut être joué par différentes personnes à différents moments et il a une responsabilité claire dans le processus. Ce sont les rôles qui exécutent les différentes étapes humaines d'un processus, pas les personnes. Une personne réelle peut interpréter plusieurs rôles dans un processus.

²¹ Cf. Figure 17.

A titre d'exemple, le diagramme de la Figure 17 montre les différents acteurs et leurs relations du processus NS-P1 du projet *Acoustic* :

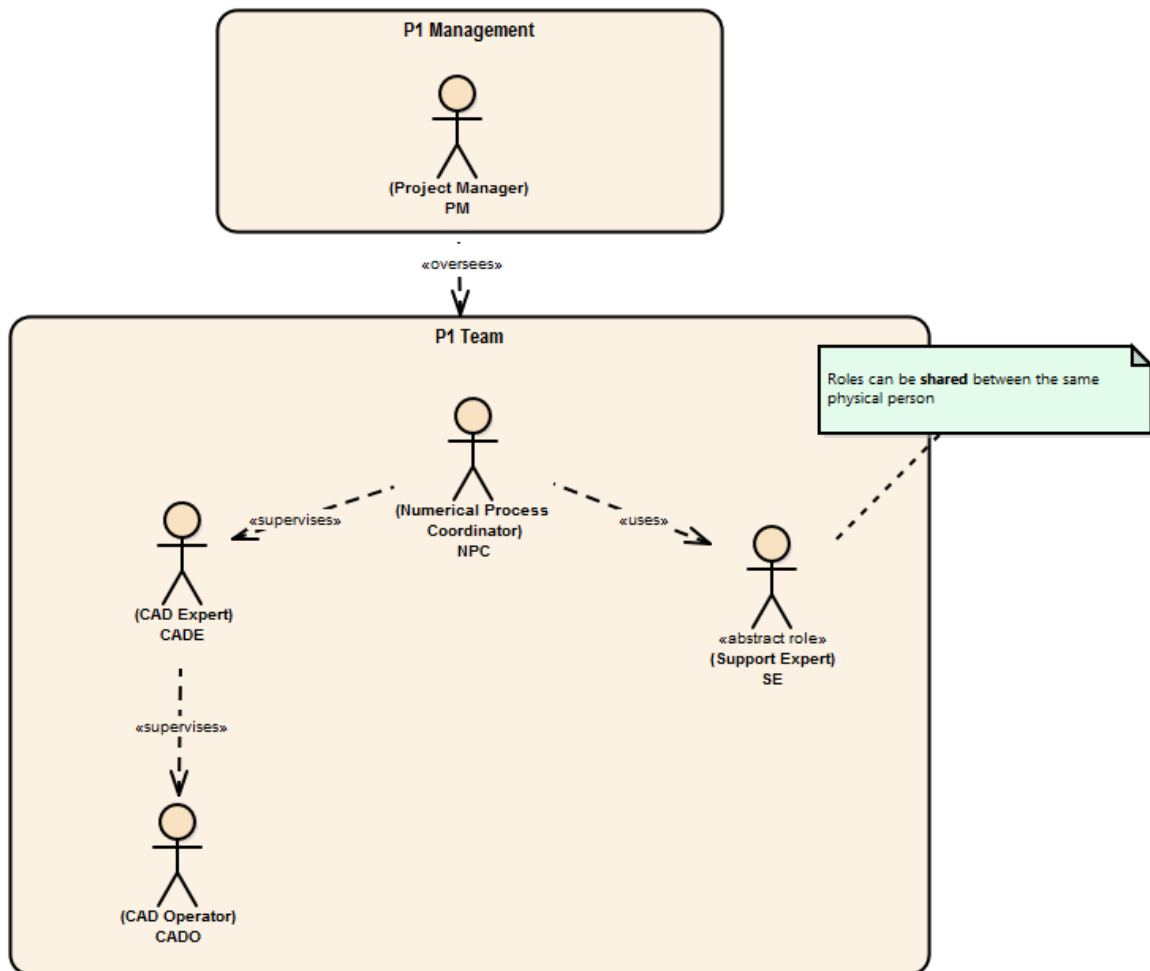


Figure 17 - Liste des rôles du processus NS-P1 sous la forme d'acteurs UML

D'expérience, j'ai constaté que le fait de se poser et de lister les rôles et leurs responsabilités dans un processus permet d'un part de se mettre d'accord sur le nom des rôles et d'autre part de remettre en question les responsabilités existantes pour les améliorer.

Ayant une cartographie comme point de départ, la prochaine étape est la définition des **scénarios** pour chaque rôle dans le processus formalisé sous la forme de **cas d'utilisation** UML selon les principes de la méthodologie ICONIX²². Chaque cas d'utilisation est identifié à partir de cette cartographie puis regroupé par rôle dans des packages UML.

Le diagramme de la Figure 18 liste les cas d'utilisation dans de packages pour chaque rôle (ou acteur) du processus NS-P1 du projet *Acoustic*.

²² Cf. section III.3.2 « Méthodologie ICONIX »

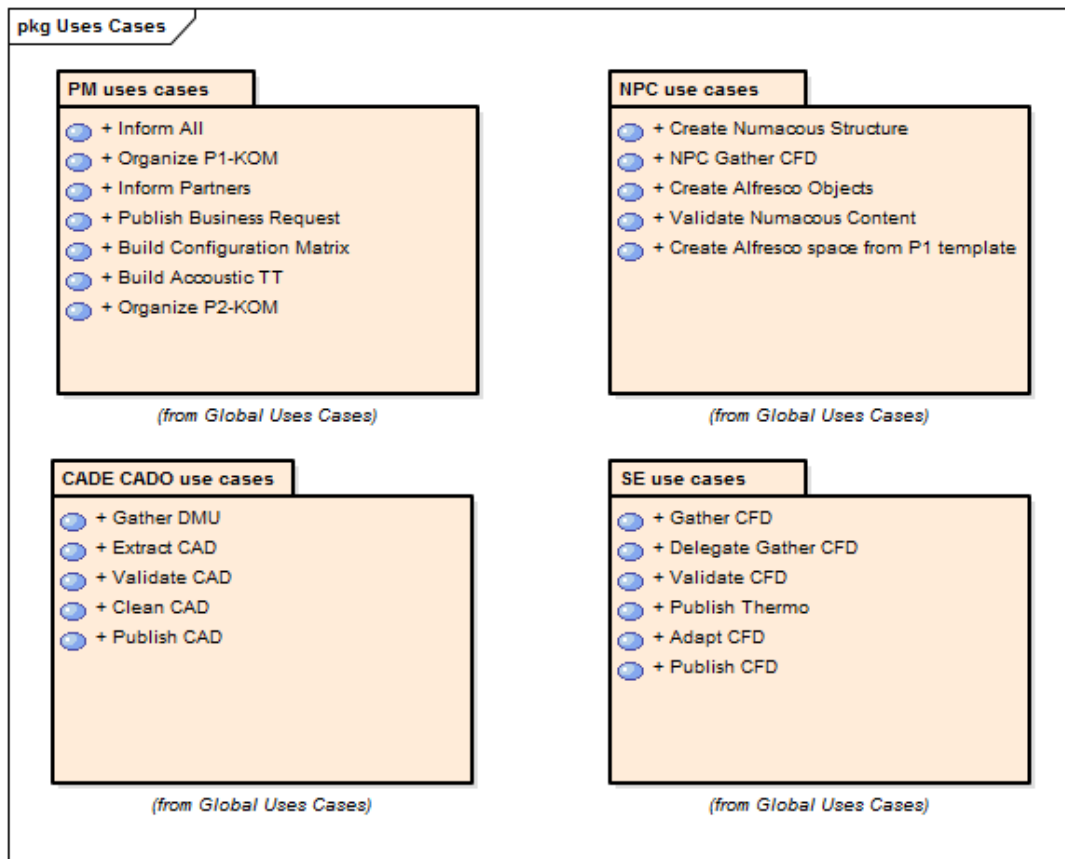


Figure 18 - Regroupement des cas d'utilisation par acteur du processus NS-P1

Le fait de modéliser le domaine, les acteurs et les cas d'utilisations permet aux responsables du processus de constater des incohérences et des points d'amélioration dans le processus décrit au niveau de la cartographie initiale. Cette prise de conscience a permis d'optimiser l'intervention de chaque rôle. A cette étape nous avons une meilleure vision du processus qui aboutira et nous obtenons une meilleure compréhension des points suivants :

- **Responsabilités** : qui fait quoi ?
- **Allocation de ressources** : qui utilise quoi ?
- **Relations** : qui communique avec qui ?
- **Circulation de l'information** : quels sont les documents entrants et sortants ?
- **Identification des chemins critiques**

Un des critères définissant un processus métier est l'information entrante et sortante. Le domaine est enrichi avec tous les documents pouvant circuler pendant l'exécution du processus. Un des points forts de la méthodologie ICONIX est l'évolution du domaine en classes métier ayant des attributs. Afin de pouvoir instancier un processus sur une solution technologique, chaque source de données (ou d'information) doit être modélisée avec les différentes propriétés qui la décrit sous la forme de modèle de données.

Un **modèle de données** décrit une information (ou propriété) qui circule pendant l'exécution d'une instance d'un processus. Cela peut être le nom précis de l'instance, sa description, les rôles qui sont affectés pour traiter une tâche, la date attendue pour terminer une tâche, l'initiateur d'une tâche, des documents liés, le résultat d'une condition et ainsi de suite. Il décrit principalement quelle est l'information requise (ou entrant) par un processus métier et quelle est l'information qui y circule. Pendant la phase d'exécution de

processus ces données deviendront par la suite pour chaque instance : des **variables d'exécution** et des **variables de processus**.

La Figure 19 illustre l'évolution d'une classe de domaine et modèle de données avec l'ajout de propriétés et la précision de la relation entre le processus P1 et ses attachements documentaires.

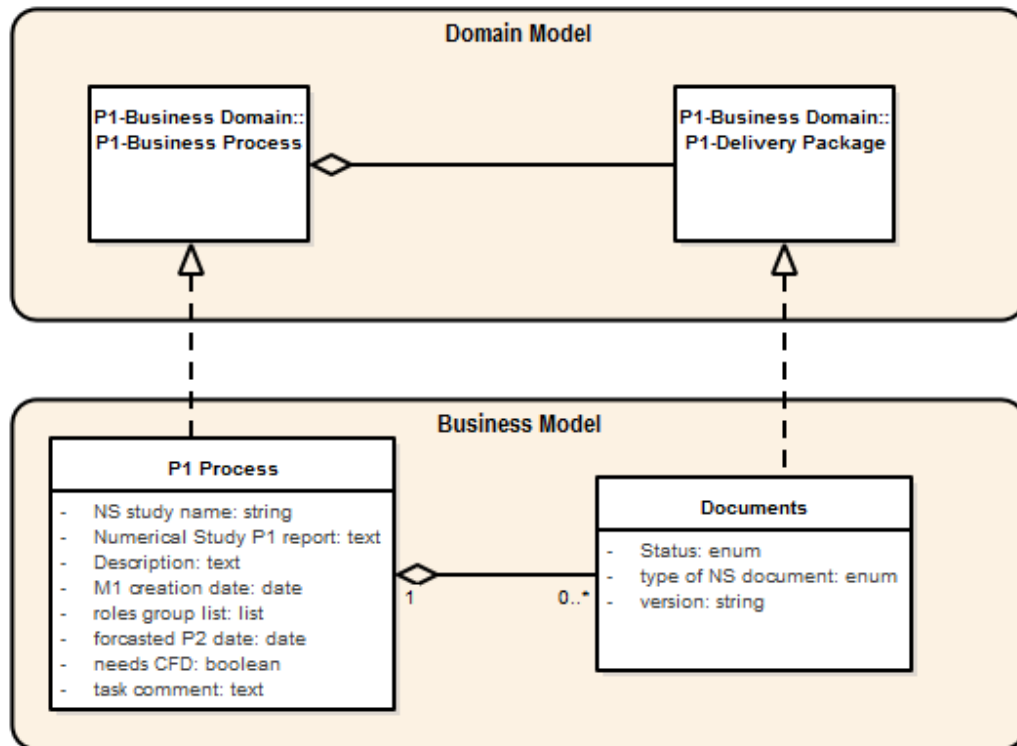


Figure 19 - Modèle de données du processus NS-P1

Pendant la réalisation du projet *Acoustic*, cette modélisation, une fois de plus, a donné l'occasion d'améliorer le niveau de qualité de l'information circulant au cours de l'exécution d'une étude numérique par l'ajout de propriétés qui n'existaient pas auparavant.

Ma démarche a été de modéliser ces éléments sous la forme de modèles informatiques avec l'outil de modélisation UML Enterprise Architect. Même si l'utilisation de document Excel, fichier texte ou autres permet d'arriver à un résultat similaire. Les modèles informatiques ont une valeur ajoutée forte au détriment des capacités requises pour les réaliser. La sous-section suivante présente la formalisation de la cartographie en modèle de processus avec l'utilisation de la syntaxe BPMN 2.0.

La démarche de modélisation est souvent aperçue comme étant compliquée par le client s'il n'y participe pas activement, puisque sa lecture n'est pas naturelle pour un non initié. Le fait de réaliser les premières ébauches de diagrammes en séance de travail permet un apprentissage de sa lecture.

Afin d'initier cette démarche de modélisation du domaine, des acteurs, des cas d'utilisation et du modèle de données, l'utilisation du diagramme de *Eriksson-Penker* s'avère utile. C'est un bon point de départ si une cartographie client n'a pas encore été faite car il permet de se focaliser sur les éléments qui sont utilisés par le processus sans pour autant rentrer

brusquement dans les détails du processus. Dans tous les cas c'est un support de modélisation efficace pour définir la modélisation des données.

III.3.3.3 Modèles de processus en BPMN 2.0

Suite à la production de la cartographie d'un processus métier par la phase d'étude préliminaire nous devons construire un modèle de processus servant de référence aux prochaines étapes. Nous allons voir dans cette sous-section de quelle manière nous pouvons élaborer des modèles de processus BPMN 2.0.

Selon la discipline BPM, la modélisation de processus pourrait être résumée en une méthodologie qui permet de parvenir à la représentation graphique d'une série d'événements, d'activités et d'actions mises en séquence au moyen de diagrammes servant de référentiels.

La transposition d'un processus métier en modèles de processus BPMN 2.0 est une démarche fondamentale et nécessite la mise à disposition de standards à suivre afin d'obtenir une cohérence en se basant sur un langage commun pour décrire visuellement un processus de bout en bout.

Contrairement à la vue « à plat » de la cartographie informelle d'un processus, les modèles de processus sont des fichiers dynamiques. La création de modèles de processus doit se faire avec un outil de modélisation²³. L'utilisation d'un outil de modélisation permet principalement d'organiser et de lier les différents diagrammes, de définir la hiérarchie parent/enfant des processus, de mettre à jour des modèles rapidement et de générer et maintenir rapidement la documentation. Une modification faite sur un modèle est immédiatement reportée sur tous les diagrammes l'utilisant. Les relations permettent d'assurer une traçabilité des éléments. Un autre atout non négligeable sur l'utilisation d'un outil de modélisation est de pouvoir évaluer la performance d'un processus métier par le biais de l'exécution de simulations. En pratique je n'ai pas encore trouvé d'outil de simulation BPMN 2.0 satisfaisant, cette pratique reste manuelle, mais je suis persuadé de sa valeur ajoutée car je l'utilise activement dans la modélisation machines à état. Je vois la simulation comme l'insertion de jetons dans le processus pour constater comment il se comporte afin de le rendre robuste.

La première étape de la construction d'un modèle de processus est la définition du standard de modélisation à adopter. Il existe une variété de standards et ce mémoire présente l'utilisation du standard BPMN 2.0 pour les raisons évoquées dans le chapitre précédent²⁴.

²³ Ici Enterprise Architect.

²⁴ Cf. section 0 "un gage d'efficacité et permet une facilitation de l'apprentissage. Elle devient un référentiel et sert de base pour toutes communications de processus. Ce référentiel est structuré et documenté et il faut être certain que la méthode de modélisation soit correcte dans le sens où elle est applicable, adoptée et permet d'obtenir des résultats exploitables. Ensuite il faut s'assurer que la méthode ait été appliquée dans les modèles et que le processus soit compris.

Je vous ai présenté qu'un processus métier est une séquence de tâches dans une organisation avec l'objectif de faire une activité à valeur ajoutée. Un processus en BPMN peut décrire différents niveaux de processus métiers : du processus haut-niveau d'une entreprise à un processus unitaire fait par un seul rôle. Chaque processus décrit en BPMN 2.0 peut être réutilisé individuellement pour atteindre des objectifs différents selon la situation.

Chaque activité de la cartographie d'un processus doit être transposée en élément BPMN 2.0. La spécification BPMN 2.0 prévoit plusieurs types de tâches (ou *Tasks*), en réalité, d'après mon expérience, seulement ceux de la Figure 20 ci-dessous sont utilisés par la majorité des moteurs de workflow. Pourquoi cette limitation ? A mon sens c'est plus efficace de viser l'objectif final d'automatisation et d'éviter de se polluer avec des typologies de *Tasks* qui ne seront pas gérées par la suite car complexes à exécuter. Un des objectifs de la syntaxe BPMN 2.0 étant son exécution dans un moteur de workflow.

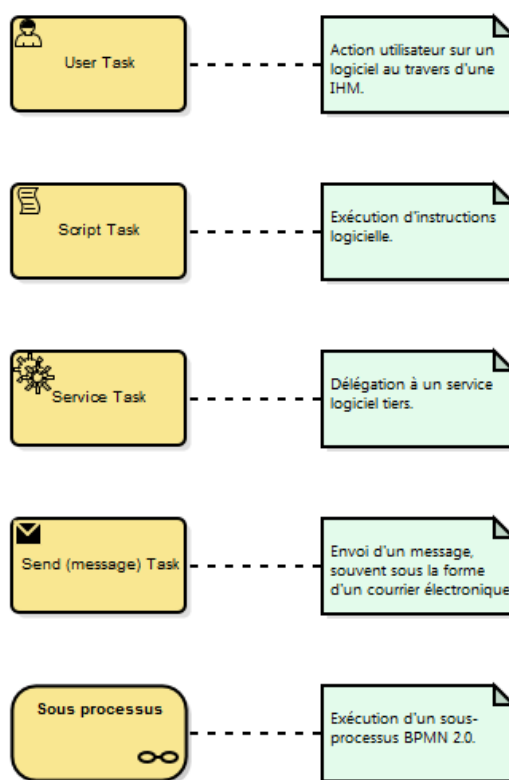


Figure 20 - Types de *Task* BPMN 2.0 principalement utilisés.

Afin de pouvoir transposer un processus au format BPMN 2.0 je vais commencer par exposer comment ce formalisme décrit les processus et quels sont les apports de cette notation.

La réussite de la modélisation de processus métiers dépend essentiellement des choix méthodologies et techniques. Cette partie terminera par une méthodologie qui a été mise en œuvre intégrant la collaboration des parties prenantes durant le cycle de vie du projet.

Après analyse de la cartographie initiale, la prochaine étape est l'identification des **activités humaines**, c'est à dire, chaque activité qu'un utilisateur doit effectuer pour avancer dans le processus, par exemple valider un document, publier un document, etc. En BPMN 2.0 ces activités deviennent des *Tasks* de type *User* (ou utilisateur). Un *User Task*, par définition, implique une interaction avec un système logiciel par le biais de IHM.

Certaines activités dans un processus ne seront jamais exécutées par des humains et seront automatiques, la prochaine étape est la découverte des **activités machine (ou automatiques)**. Ce sont toutes les activités utilisant un programme pour s'exécuter automatiquement avec des lignes d'instructions ou l'appel à des services tiers par le biais d'interfaces applicatives de type REST ou SOAP.

Les activités exécutées par des machines sont les : **Script Tasks, Service Tasks et Message Tasks**.

Le langage de programmation avec laquelle les instructions logiques du script sont écrites est fortement lié avec le moteur de workflow, s'il y en a un, qui l'exécute. Si le processus n'est pas exécuté par un moteur de workflow, les tâches automatiques doivent tout de même être indiquées puisqu'elles peuvent représenter une macro Excel ou un envoi de message électronique avec un outil de messagerie par exemple. A ce stade de la formalisation BPMN 2.0 il n'est pas nécessaire de préciser comment ces programmes seront implémentés et quelle technologie sera utilisée. Par-contre les services doivent être identifiés. Nous devons nous positionner d'un point de vue de la conception dans le domaine du problème et pas encore dans la solution technologie dans ses détails. Ce document présentera plus loin²⁵ comment exploiter la syntaxe du langage XML-BPMN 2.0 sous-jacente pour implémenter ces instructions automatiques.

Ensuite chaque **branchement** doit être identifié à son tour. Un branchement est une passerelle permettant de modéliser les décisions, fusions, fourches et jointures d'un processus. En langage BPMN 2.0 les branchements se nomment *Gateway*. La cartographie est analysée afin d'identifier les branchements exclusifs et les activités pouvant s'exécuter en parallèle.

Les branchements exclusifs sont modélisés avec un *Exclusive Gateway* et les groupes d'activités exécutées en parallèle avec des couples *Parallel Gateway / Inclusive Gateway*. Comme pour les *Tasks*, seulement quelques branchements sont réellement utilisés par les moteurs de workflows. La liste de ces branchements est présentée dans la Figure 21.

²⁵ Cf. chapitre 0 “ Anatomie de la représentation XML-BPMN 2.0”

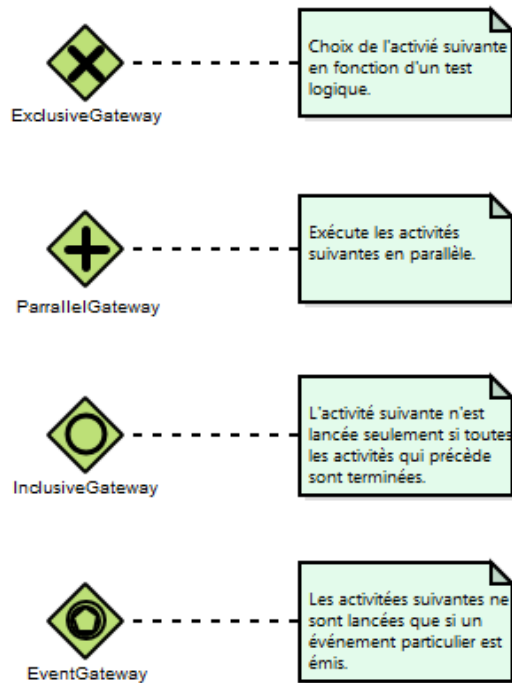


Figure 21 - Types des branchements BPMN 2.0 principalement utilisés.

Le sens du flux après un branchement exclusif est déterminé par des règles de logique qui répondent à une question : le document est-il valide ? La demande est acceptable ? Le montant de la commande est-elle supérieure à un seuil ? ... Sur chaque association de type séquence sortante du branchement on notera la réponse à cette question, afin de faciliter la lecture. N'oublions pas qu'un diagramme BPMN 2.0 a pour vocation d'être validé par des non-techniques et l'ingénieur de processus doit faire preuve d'empathie et de simplicité dans ses notations textuelles. Nous allons découvrir par la suite que sous le diagramme BPMN se situe un langage XML extensible qui portera la logique technique derrière les branchements sous la forme de programmes exécutables.

La série de figures suivantes montre quelques patrons de branchements habituels identifiés sur différents projets, appliqués sur le projet *Acoustic*.

Les boucles permettent de répéter une ou plusieurs *Tasks*.

La Figure 22 montre une boucle type. Cette boucle est constituée d'un branchement exclusif (ou *Exclusive Gateway*) qui retourne sur la tâche utilisateur « *SE : Gather CFD* ». Tant que ni le *PM* et le *SE*²⁶ valide le contenu du document *CFD*, le document est refusé et une notification est faite avec la tâche automatique « *Inform All* ». Nous pouvons observer l'utilisation d'un branchement parallèle (ou *Parallel Gateway*) pour représenter la validation conjointe du *PM* et du *SE*.

²⁶ Le rôle *PM* est l'abréviation de *Process Manager*, et le rôle *SE* est l'abréviation de *Support Expert*.

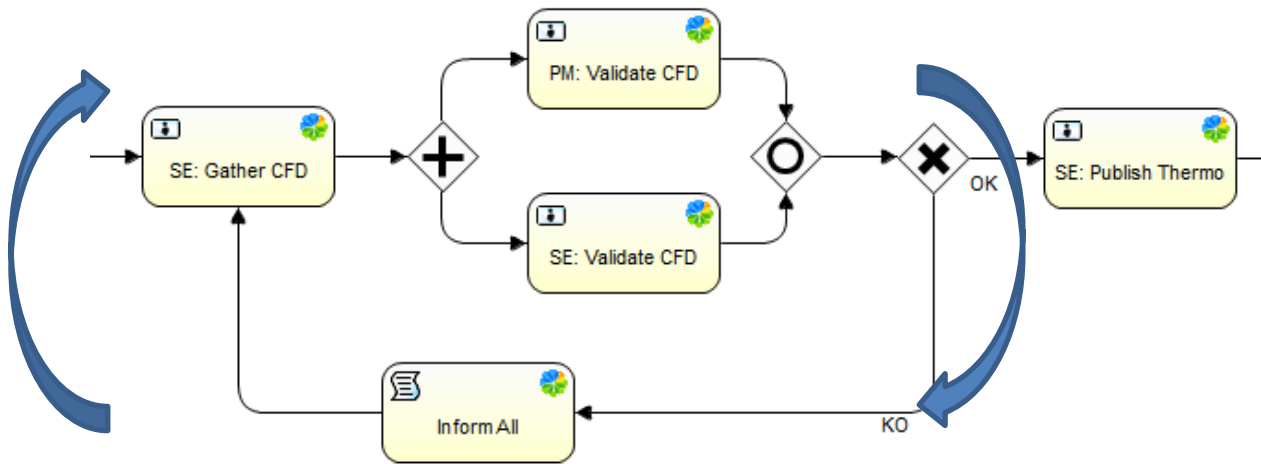


Figure 22 - Pattern BPMN 2.0 de boucles.

L'exécution d'activités en parallèle, tant que toutes les activités entre un branchement *Parallel Gateway* et *Inclusive Gateway* ne sont pas terminées, l'activité suivante ne sera pas instanciée. L'application de ce pattern permet de représenter les cas où plusieurs acteurs doivent réaliser des tâches et tant qu'elles ne sont pas terminées nous ne pouvons pas procéder à la suivante. Ce pattern est représenté dans le diagramme de la Figure 23.

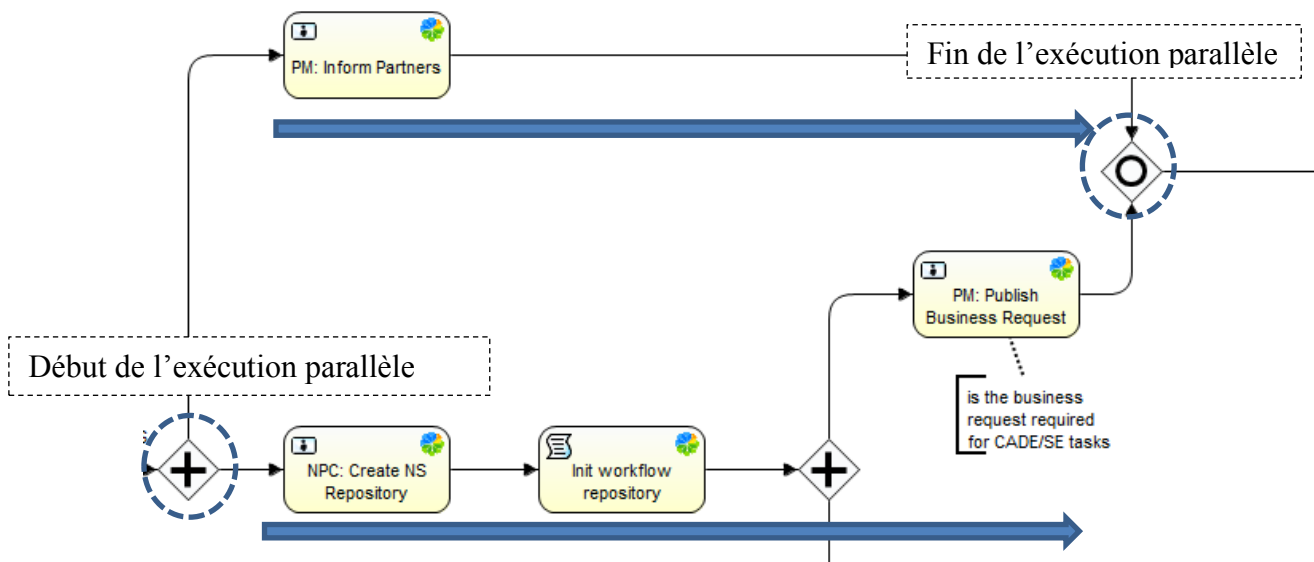


Figure 23 - Pattern BPMN 2.0 d'exécution parallèle.

L'interruption d'un processus est un pattern où selon certaines conditions le processus entier doit s'arrêter. Ce pattern de décision simple d'interruption de processus est représenté dans la Figure 24.

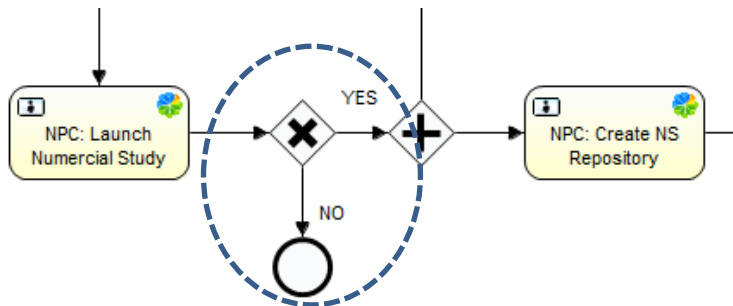


Figure 24 - Patterns BPMN 2.0 d'interruption.

Finally, we identify the or the **événements déclencheurs du processus** and the **points d'arrêt et de fin**. They are formalized with the BPMN 2.0 *Start Event* and *End Event* presented in the following figure.²⁷

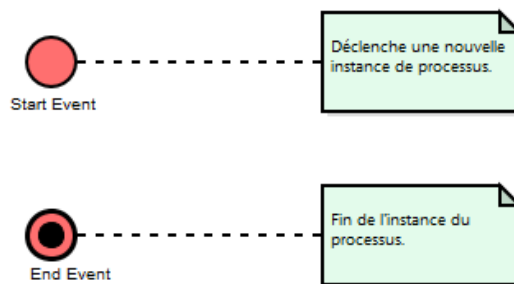


Figure 25 - Types d'événements BPMN 2.0 principalement utilisés.

This transposition is sufficient to be able to complete the definition of process by a technical resource having the technological knowledge of the target solution. Then it allows to validate in details the process from a business point of view.

A domain has been described allowing to frame the vocabulary of the IT project according to the ICONIX methodology. Each activity name must be formulated with the support of this vocabulary, ideally this name is directly issued from the name of use case to a particularity, I find that it is more comprehensible to add the abbreviation of the actor before the name of an activity of type *User Task*. Even if the standard BPMN 2.0 allows to regroup the activities in lanes (or *Lanes* in English).²⁸

²⁷ D'autres événements sont utilisés par le moteur de workflow Activiti, mais n'ont pas été utilisés dans les deux projets présentés dans ce mémoire.

²⁸ Cette pratique est issue d'un problème rencontré avec le moteur de workflow Activiti expliqué dans la section IV.4.1 « Intégration dans Activiti Engine »

Afin d'identifier clairement une activité indépendamment de son nom, un identifiant unique (ou *UID*) doit lui être associé. Les outils de modélisation BPMN 2.0 affectent automatiquement cet identifiant et selon la convention de nommage du projet il peut être adapté. Cela permet d'associer puis de référencer les activités directement à l'intérieur du langage XML sous-jacent.

La représentation graphique d'un processus sous la forme de diagramme peut très rapidement avoir une lecture difficile si le nombre d'éléments qui y figurent est important. Un processus très complexe peut être reformulé en segments de sous-processus avec plusieurs diagrammes. Il est nécessaire d'être vigilant sur ce point car si le processus a le potentiel d'être instancié par un moteur de workflow nous avons un risque d'être techniquement limité. La structuration des processus est d'autant plus importante que si un processus dépend d'un autre, ce qui a été le cas dans le projet *Acoustic* avec les sous-processus P1, P2 et P3 du processus NS. Je pense que nous pouvons aborder cette structuration voire factorisation de processus comme le paradigme de programmation modulaire où chaque composant logiciel est interchangeable et réutilisable.

Une focalisation sur la modélisation de la représentation graphique BPMN 2.0 a été faite dans cette section. Le processus est maintenant suffisamment formalisé et précis pour être implémenté et exécuté humainement et/ou dans des logiciels puis maintenu tout en respectant le cycle BPM.

Dans les annexes 2 et 3 vous pouvez consulter le processus *Acoustic* NS-P1 avant et après formalisation en BPMN 2.0.

IV Instanciation de processus

Suite à la transposition, la modélisation est suffisamment aboutie pour essayer d'intégrer le processus dans la solution logicielle visée. Nous allons découvrir comment la définition de processus doit être adaptée pour qu'il puisse être exécutable. Le moteur de workflow sera capable d'interpréter cette définition de processus adapté puis de l'instancier pour exécution.

La Figure 26 ci-après met en avant (cadre rouge) où se situe cette partie du mémoire dans les étapes de transposition d'un processus métier. Une synthèse du processus complet de la méthodologie est présentée en annexe 1.

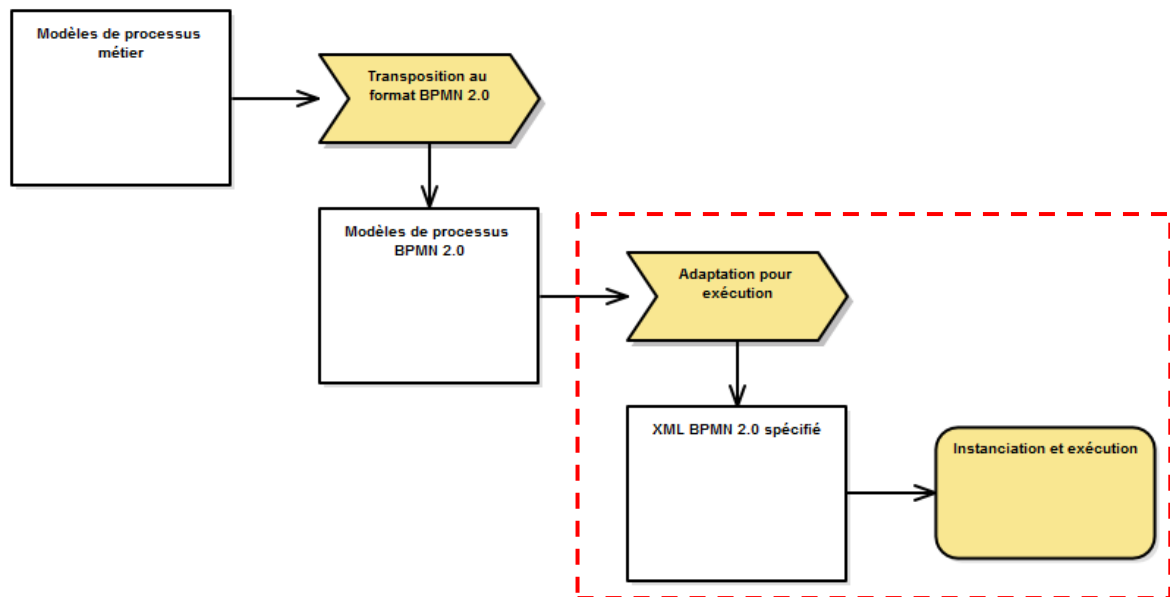


Figure 26 - Etapes de transposition d'un processus métier, focus instanciation et exécution.

Différentes solutions compatibles avec la spécification BPMN 2.0 existent aujourd'hui pour instancier et exécuter un processus et dans le cadre de ce projet je me suis intéressé à deux moteurs de workflow *open source* en Java :

- **Activiti** : plate-forme de modélisation et d'exécution de processus soutenue par Alfresco créée en 2010 s'adressant essentiellement à des développeurs. Cet outil permet une grande flexibilité mais est peu accessible aux utilisateurs autres que des développeurs par manque d'IHM de gestion de processus.
- **Bonitasoft** : plate-forme de modélisation et d'exécution de processus mature proposant des IHM de gestion de processus. La définition de processus et des formulaires se font exclusivement de manière graphique et le code est généré.

Bonitasoft est une solution permettant une mise en œuvre de processus rapidement et permet de se connecter avec de nombreux services tiers nativement. Activiti est une solution qui nécessite un plus grand effort d'intégration mais sa flexibilité permet des intégrations spécifiques dans des systèmes d'information existantes. Contrairement à Activiti, avec Bonitasoft les développeurs n'ont pas un contrôle total sur le code de l'exécution de processus. Par-contre les responsables métiers ont plus d'autonomie avec Bonitasoft.

Activiti a été retenue pour le projet *Acoustic* par sa flexibilité, la possibilité de contrôler intégralement le code de la définition du processus et son intégration native dans Alfresco.

La section III.1 a présenté que le standard BPMN 2.0 vise à la fois les humains et les machines par le biais de sa représentation XML. Le chapitre suivant va vous décrire une synthèse de l'anatomie de la partie sous-jacente XML d'un diagramme BPMN2.0 afin d'identifier comment rendre une définition de processus métier décrite avec la syntaxe BPMN 2.0 exécutable par **spécialisation** de son XML.

IV.1 Anatomie de la représentation XML-BPMN 2.0

Techniquement le XML de la définition d'un processus est organisé en deux parties : une section **descriptive** du processus et une section **représentation graphique**. La section représentation graphique n'est pas obligatoire pour que le processus décrit puisse être exécuté, c'est la partie descriptive qui sera interprétée par le moteur de workflow. Je vais vous montrer comment fonctionne rapidement ces deux sections.

IV.1.1 La définition de processus

La partie descriptive du processus est appelé la **définition de processus** (ou *process definition* en anglais) et présente la description du séquençement des étapes de ce processus et éventuellement de ses sous-processus. Chaque élément XML de cette définition représente les métadonnées d'un élément graphique du processus BPMN 2.0 : son type, identifiant unique, son nom, d'éventuels marqueurs, documentation, relation avec d'autres éléments²⁹.

Examinons le XML de cette section :

```
<process id="exemple" name="Exemple de processus" isExecutable="true">
  <startEvent id="start"></startEvent>
  <sequenceFlow id="flow1" sourceRef="start" targetRef="validDocumentTask"></sequenceFlow>
  <userTask id="validDocumentTask" name="Validation de document">
    <extensionElements></extensionElements>
  </userTask>
  <sequenceFlow id="flow2" sourceRef="validDocumentTask" targetRef="verifierTerminee"></sequenceFlow>
  <userTask id="verifierTacheTerminee" name="Vérification tâche validation terminée.">
    <documentation>Vérification que la tâche de validation a été faite.</documentation>
    <extensionElements></extensionElements>
  </userTask>
  <sequenceFlow id="flow3" sourceRef="verifierTacheTerminee" targetRef="fin"></sequenceFlow>
  <endEvent id="fin"></endEvent>
</process>
```

Le moteur d'exécution de processus est responsable d'interpréter cette définition pour en créer des instances. Comment ? Nous constatons rapidement que cette définition XML est insuffisante puisqu'il n'indique pas quels sont les rôles responsables d'exécuter les *User Tasks* , ni quelle est la définition des différentes IHM, quelles sont les variables d'exécution du processus, quels sont les scripts, etc.

La spécification BPMN prévoit des points d'extensions au niveau de chaque élément **extensionElements** et au niveau du *namespace*³⁰ XML. Ce qui permet d'adapter le

²⁹ Namespace XML BPMN 2.0 : <http://www.omg.org/spec/BPMN/20100524/MODEL>

³⁰ Terme anglais pour : espace de noms

XML pour le moteur de workflow cible. Une définition de processus BPMN doit être spécialisée pour être exécutée par un moteur de workflow.

IV.1.2 La représentation graphique de processus

La représentation graphique d'un diagramme de processus est décrite dans la section *bpmndi* du XML, la spécification OMG de BPMN 2.0 prévoit un format d'échanges générique de données de représentation de diagrammes BPMN appelé **BPMN DI**³¹ qui gère les aspects graphiques suivants : type formes graphique (*BPMNShape*), plans graphiques (*BPMNPlane*), bordures graphiques (*BPMNEdge*), coordonnées, largeur, hauteur, etc. C'est une syntaxe abstraite que chaque outil de représentation graphique interprète en corrélation avec la section descriptive, pour réaliser l'affichage du diagramme final.

Examinons le XML cette section :

```
<bpmndi:BPMNDiagram id="BPMNDiagram_exemple">
  <bpmndi:BPMNPlane bpmnElement="exemple" id="BPMNPlane_exemple">
    <bpmndi:BPMNShape bpmnElement="start" id="BPMNShape_start">
      <omgdc:Bounds height="35.0" width="35.0" x="145.0" y="200.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="validationDocumentTask" id="BPMNShape_validationDocumentTask">
      <omgdc:Bounds height="55.0" width="105.0" x="220.0" y="190.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="verifierTacheTerminee" id="BPMNShape_verifierTacheTerminee">
      <omgdc:Bounds height="55.0" width="105.0" x="365.0" y="190.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNShape bpmnElement="fin" id="BPMNShape_fin">
      <omgdc:Bounds height="35.0" width="35.0" x="510.0" y="200.0"></omgdc:Bounds>
    </bpmndi:BPMNShape>
    <bpmndi:BPMNEdge bpmnElement="flow1" id="BPMNEdge_flow1">
      <omgdi:waypoint x="180.0" y="217.0"></omgdi:waypoint>
      <omgdi:waypoint x="220.0" y="217.0"></omgdi:waypoint>
    </bpmndi:BPMNEdge>
    <bpmndi:BPMNEdge bpmnElement="flow2" id="BPMNEdge_flow2">
      <omgdi:waypoint x="325.0" y="217.0"></omgdi:waypoint>
      <omgdi:waypoint x="365.0" y="217.0"></omgdi:waypoint>
    </bpmndi:BPMNEdge>
    <bpmndi:BPMNEdge bpmnElement="flow3" id="BPMNEdge_flow3">
      <omgdi:waypoint x="470.0" y="217.0"></omgdi:waypoint>
      <omgdi:waypoint x="510.0" y="217.0"></omgdi:waypoint>
    </bpmndi:BPMNEdge>
  </bpmndi:BPMNPlane>
</bpmndi:BPMNDiagram>
```

Cette représentation graphique est fortement liée avec la définition de processus et est maintenue à jour automatiquement par les outils de modélisation.

IV.1.3 Interopérabilité des processus basé sur le standard XML

Un premier niveau d'interopérabilité est possible avec cette représentation XML d'un processus. Le standard BPMN 2.0 est fondé sur ce concept dont les points forts sont listés ci-après :

- Le XML est un standard

³¹ Namespace XML BPMN DI : <http://www.omg.org/spec/BPMN/20100524/DI>

- L'OMG fournit des schémas de transformation XML de et vers le format standard de modèles informatiques le XMI.
- Les outils de modélisation qui implémentent le schéma XML BPMN offrent la possibilité de modéliser avec le standard BPMN 2.0
- Le XML est extensible pour véhiculer des informations spécifiques un moteur de workflow.

Comment spécialiser la définition XML d'un processus ?

- Par **point d'extension** : En ajoutant des éléments enfants à l'élément parent XML-BPMN 2.0 **extensionElements** avec un espace de nom unique.
- Par **attributs spécifiques** : En ajoutant des attributs spécifiques aux éléments XML-BPMN 2.0 existants avec un espace de nom unique.

Chaque éditeur de moteur de workflow compatible avec la spécification BPMN 2.0 se doit de fournir la spécification XML des points d'extensions utilisées et un *namespace* unique pour éviter des collisions.

Le chapitre suivant présente le projet *Acoustic* cité à plusieurs reprises dans ce document. Ensuite la méthodologie pour implémenter un processus dans le moteur de workflow Activiti afin de pouvoir l'instancier puis l'exécuter vous sera présentée.

IV.2 Cas 1 : Airbus Acoustic Workflows



IV.2.1 Processus d'études numériques

J'ai été positionné par la société d'ingénierie en informatique **X2i**, spécialisé dans le commerce électronique, dans laquelle je suis salarié, sur un projet d'informatisation de processus humain pour une section d'**Airbus** dédiée à l'étude numérique de phénomènes acoustiques : **Numerical Methods Group Acoustics & Environment**. Ce projet a eu pour objectif la mise en œuvre d'un prototype démontrant la faisabilité de la transformation d'un processus humain en modèles de processus informatiques formalisés en BPMN 2.0 et exécutés par une application de gestion électronique de contenus utilisée par Airbus : Alfresco.

Ce processus humain est référencé comme l'activité **EEA6 Business Process Numerical Study** et le projet a été référencé sous le nom de **Numerical Study (EEA6) Process P1 Mockup** par Airbus.

X2i a été sollicité pour la qualité de ses ingénieurs et son expertise dans les domaines de la modélisation de processus en BPMN 2.0 et la customisation de l'application Alfresco. Etant initié dans la norme BPMN 2.0 par mon expérience en l'informatisation de processus métiers dans le secteur du commerce électronique et ayant un niveau bilingue en anglais, j'ai été introduit pour traiter ce projet techniquement et apporter mon expertise.

L'objectif du processus métier d'entreprise **EEA6 Business Process Numerical Study** est l'étude numérique des phénomènes acoustiques de différents composants avioniques et de leur revêtement insonorisant soumis à certaines conditions et contraintes aérodynamiques, géométriques, thermiques... Une étude numérique est initiée par une demande métier appelée « Business Request » approuvée par un groupe d'analystes qui a pour objectif la livraison d'un rapport détaillé des résultats de simulations obtenus.

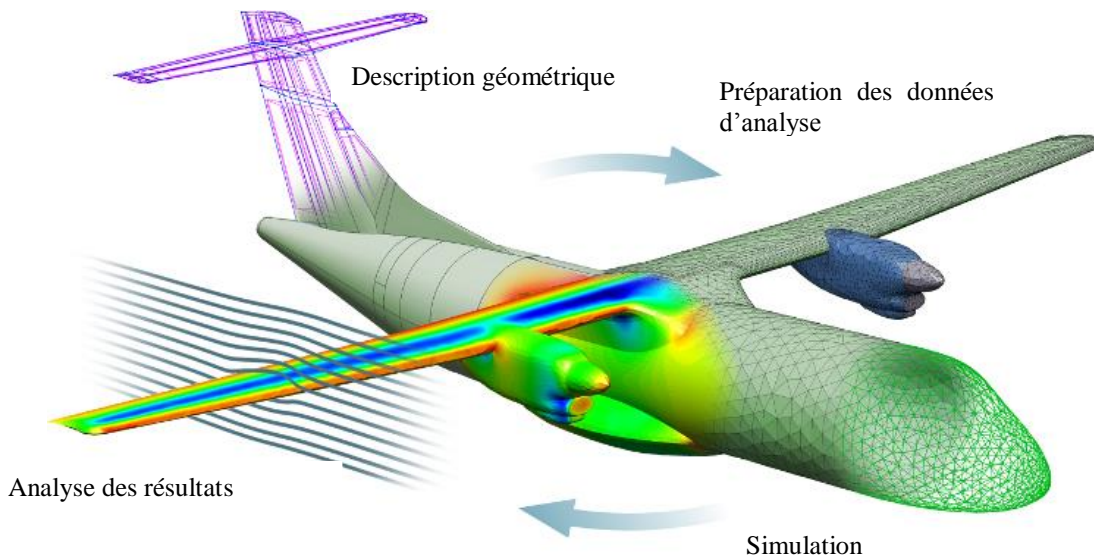


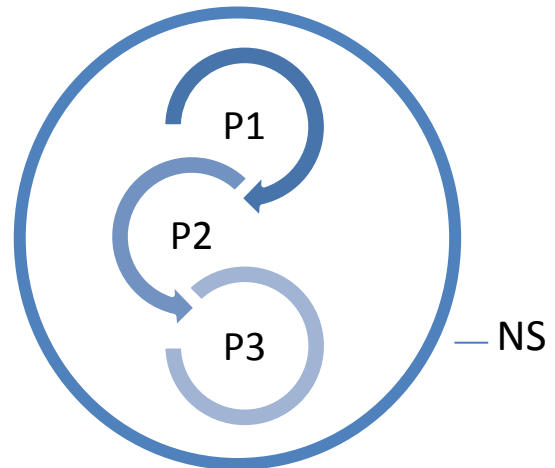
Figure 27 - Etapes d'une étude numérique avionique.

Il peut y avoir entre 50 et 100 études numériques faites indépendamment ou simultanément sur une période d'un an.

Ce processus d'analyse numérique est humain et nécessite la mobilisation et la collaboration de différents intervenants et organisations ayant des rôles et responsabilités différents. Le moyen principal d'échanges est par courrier électronique, documents écrits appels téléphoniques et réunions de travail.

Le processus *EEA6 Business Process Numerical Study* est composé par trois phases ou sous-processus, respectivement P1, P2 et P3.

- **P1** : responsable de la définition des **spécifications** indispensables pour une étude numérique.
- **P2** : responsable du **déroulement** de l'étude numérique.
- **P3** : responsable de **l'analyse et la synthèse** aboutissant à la publication du **rapport final** de l'étude numérique.



Le processus NS est un processus métier d'entreprise puisqu'il est constitué d'entrants et de sortants, a un objectif et est utilisé par des ressources. Les deux prochains diagrammes montrent la vue d'ensemble du processus NS fourni initialement par le client puis ma reformulation synthétique utilisant la notation *Eriksson-Penkner* afin de mettre en avant les entrants, sortants et les objectifs des phases P1, P2 et P3.

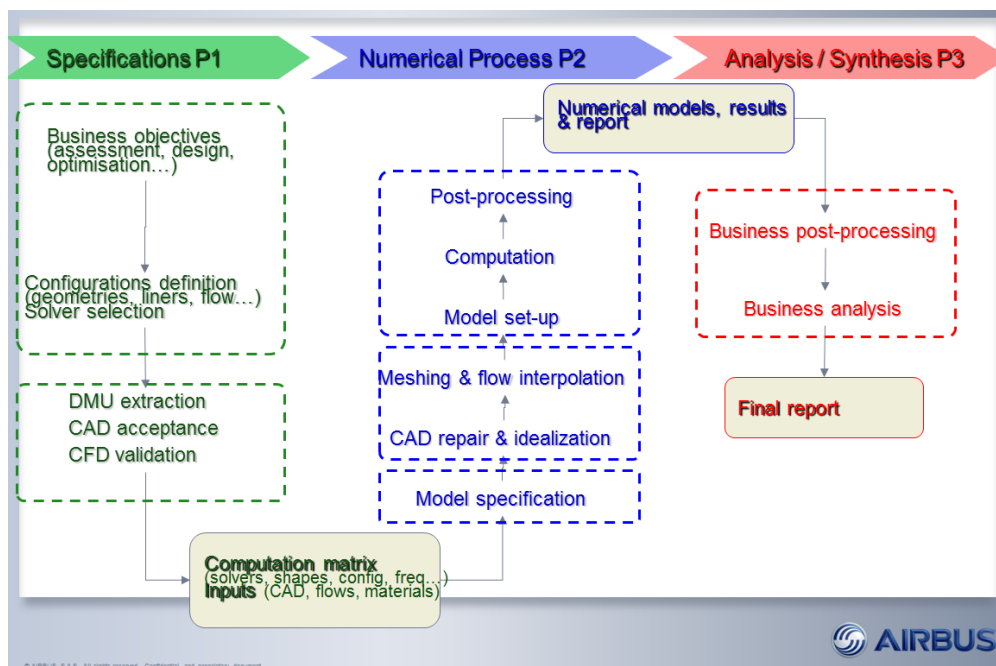


Figure 28 - Vue d'ensemble du processus NS, en anglais, des parties principales.

La Figure 29 montre le processus NS avec ses 3 sous-processus au format *Eriksson-Penker* afin de le synthétiser et faciliter sa lecture.

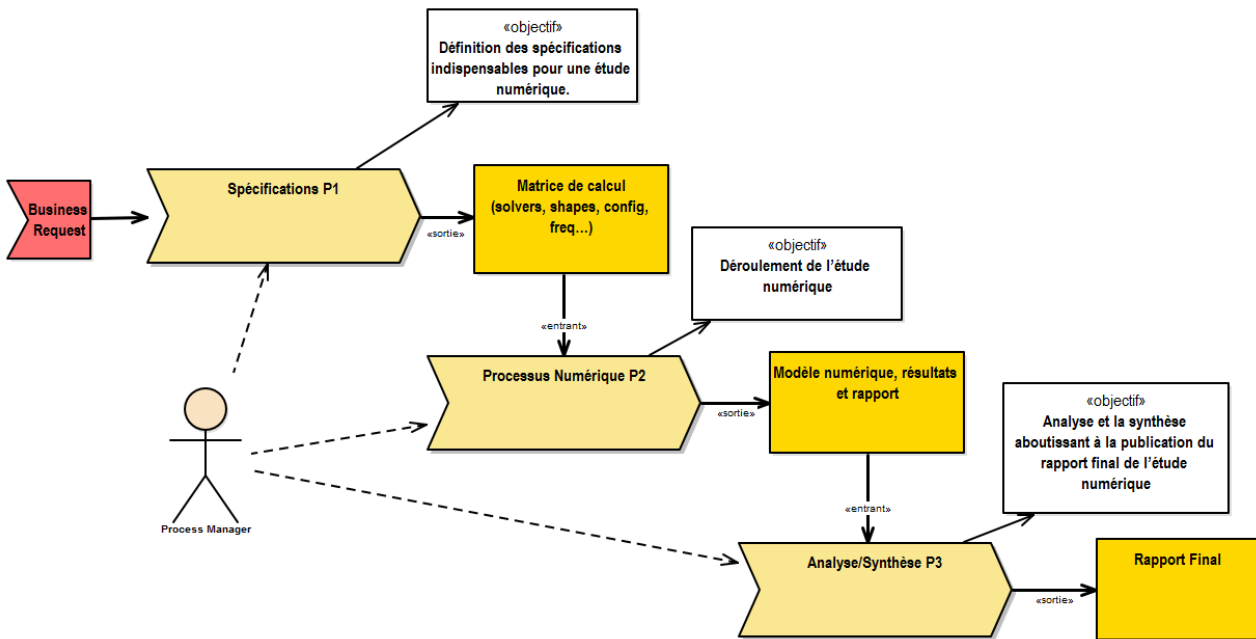


Figure 29 - Processus NS en notation Eriksson-Penker.

Le projet confié à X2i a été focalisé sur la phase P1 mais prend en compte toutes les phases du processus *NS* dans son ensemble. Je vais vous décrire sommairement le fonctionnement actuel de la phase P1.

Une demande d'étude numérique est proposée à l'équipe *Acoustic Platform*. Le responsable organise une réunion de lancement avec la participation d'experts et d'opérationnels afin de valider la faisabilité de cette demande. Il formalise la demande métier en « Business Request » puis prépare l'espace de stockage pour les différents documents techniques et informe les partenaires qu'une nouvelle étude sera lancée afin qu'ils puissent organiser la disponibilité de leurs équipes. Les opérationnels et experts rédigent les données de simulation chacun dans leur domaine d'expertise. Dès que tous les documents sont validés et fournis, une matrice de calcul est réalisée. En cas de problème tous les participants doivent être informés afin d'acter sur les solutions possibles. L'étude peut être remise en question et des décalages peuvent avoir lieu chez les partenaires. Enfin tous les documents de l'étude sont validés et transmis à l'équipe de la phase suivante.

Au cours de la phase de spécification P1 plusieurs variables et contraintes de l'étude numérique sont définis : les objectifs métiers (évaluations, conceptions, optimisations...), la définition des paramètres de simulation (géométries, « liners », flux...) et la sélection du solveur numérique adapté par la manipulation de différents documents (DMU, CAD, CFD...). Une matrice de calcul est construite puis incluse dans le paquet de livraison qui sera utilisé en tant qu'entrant pour la phase P2.

L'objectif principal de la phase P1 est l'anticipation des besoins de la phase P2, éliminant les éléments inutiles, maîtrisant les risques, créant un document *DMU* suffisamment aboutit pour construire une matrice de calcul optimisée.

La gestion et la coordination des nombreuses études numériques simultanées s'avère être compliquée.

IV.2.2 Les dysfonctionnements

Le processus NS avait été validé par l'équipe *Acoustic* en 2010 puis appliqué. A l'issue de deux années d'utilisation de ce processus humain, plusieurs dysfonctionnements sont observés par l'équipe du projet.

Ces dysfonctionnements sont listés ci-après.

- **Difficultés à partager l'information d'avancement d'un processus avec les intervenants.** Les intervenants sont multiples et un bon nombre sont dans des sociétés de service et d'expertise externes. L'anticipation et l'organisation des points de rencontres est difficile.
- **Inefficacité du temps passé à suivre les actions et les tâches.** Le suivi des tâches se fait par échanges et recherches de documents, réunions et emails. L'occupation du temps du responsable d'une étude numérique est inefficace puisqu'il doit récolter l'information afin d'agir.
- **Aucune vision globale sur le processus et incapacité à gérer les points bloquants.** Quelques tentatives de tableau de bords ont été tentées sans succès. La communication d'un point bloquant arrive tardivement. La traçabilité des activités des équipes est compliquée.
- **Incapacité d'associer du contenu technique au processus.** Ce processus met en œuvre la création, validation et circulation de nombreux documents électroniques et il s'avère être difficile de les suivre et les lier avec une étude numérique.
- **Archivage d'un processus et de son contenu impossible.** L'information est répartie entre les différents membres de l'équipe et les intervenants externes.

La difficulté majeure réside donc dans la non-centralisation de l'information et l'incapacité de gérer la coordination efficacement. L'objectif financier est d'optimiser ce processus.

IV.2.3 Objectif de la solution proposée

La solution proposée est de transformer le processus humain de la phase P1 en modèles de processus informatiques avancés, exécutés par le gestionnaire de contenu Alfresco combiné avec le moteur de workflow BPMN 2.0 Activiti, les avantages de cette solution pour le client final sont les suivants.

- Fournir une vue globale du processus **NS** aux managers.
- Fournir une traçabilité des différentes étapes du processus **NS**.
- Combiner la puissance d'Alfresco avec un moteur de workflow BPMN 2.0.
- Faciliter la maintenance de chaque instance du processus **NS**.
- Industrialisation du processus.
- Réduire les interventions, sollicitations et erreurs humaines afin de rendre le processus **NS** efficient.

Le prototype doit démontrer la faisabilité des points-clef listés ci-après.

- Affectation de tâches du workflow de la phase P1 entre différents acteurs.

- Pouvoir gérer plusieurs instances du workflow.
- Supprimer une instance du workflow.
- Suivre une instance du workflow étape par étape avec un gestionnaire de liste de tâches.

Plusieurs tentatives d'implémentation de workflow ont eu lieu et l'utilisation de la combinaison Activiti et Alfresco est un atout puisqu'elle :

- se base sur un standard de notation de processus;
- s'intègre dans l'outil de gestion de contenus électroniques permettant de centraliser l'information pour l'équipe et de minimiser l'effort de formation;
- se base technologiquement sur le langage de programmation Java connu par les équipes technique de *CIMPA* ce qui limite le risque de l'introduction d'une technologie non maîtrisée;
- propose un support collaboratif.

IV.2.4 Organisation du projet

Les étapes de ce projet ont eu lieu dans les locaux de la société X2i à Toulouse sauf la présentation et démonstration finale qui a eu lieu dans les locaux d'Airbus.

Tout au long du projet, des points de rencontres ont eu lieu afin de valider chaque étape et d'itérer sur les choix de la solution construite avec le client.

D'un point de vue organisationnel, ce projet a été divisé en 3 grandes phases successives de génie logiciel dans lesquelles j'ai activement participé en tant qu'expert, architecte et développeur.

- **Phase d'étude** permettant de définir le besoin avec l'équipe d'étude et le client puis de cadrer le projet avec le chef de projet.
- **Phase de réalisation** aboutissant en la production du livrable et de sa documentation.
- **Phase de livraison finale** aboutissant à la validation et acceptation du livrable par le client et de la signature d'un procès-verbal de recette.

L'équipe du projet chez X2i a été constitué de la manière suivante :

Chef de projet : Yannick STOLF

Référent Technique : Thierry VERCRUYSSSE

Consultant Alfresco et Java: Cyril RONSEAU

Un référent technique chez X2i a comme mission principale le maintien de la cohérence et de la validation des choix de conception, le suivi qualité, revues de code, actions correctrice, le développement, tests, le suivi des anomalies et le support de 1er niveau aux développeurs.

IV.2.4.1 Phase d'étude

La démarche que j'ai faite pour réaliser la phase d'étude repose essentiellement sur les possibilités techniques par une analyse du formalisme BPMN 2.0, du moteur de workflow Activiti et de son intégration dans Alfresco. Ayant une bonne vision de la norme BPMN

2.0 pour décrire des processus liés au commerce électronique il m'a été nécessaire de me documenter sur les possibilités d'Activiti et de son utilisation du standard BPMN 2.0. Puis de m'appuyer sur l'expertise de Cyril RONSEAU pour monter en compétences sur l'architecture d'Alfresco d'un point de vue fonctionnel et technique.

Les choix techniques ont été comparés avec des solutions similaires puis justifiés dans le contexte du projet et des domaines de compétences de l'équipe *Acoustic*. Un des objectifs de cette phase a été de définir la faisabilité du projet dans son ensemble sans rentrer dans les détails.

Le démarrage du projet a été initié par un premier groupe de travail qui a permis de définir les objectifs du projet, présenter les intervenants et leur rôle dans le projet et de réceptionner les différentes fournitures :

- Cartographie du processus NS et de ses sous-processus P1 à P3
- Des exemples de documents circulants au cours de l'étude numérique

Cette phase du projet se termine par la livraison d'un chiffrage des différentes étapes du projet au chef de projet, d'un document d'architecture globale et d'une approche pour l'équipe de production X2i.

IV.2.4.2 Phase de réalisation

Chez X2i la méthodologie organisationnelle de phase de réalisation est particulière car adaptée aux besoins de rapidité, qualité et agilité des projets de commerce électronique.

La phase de réalisation est une succession de cycles itératifs courts entre les étapes d'analyse, de conception, de réalisation et de livraison pour validation. Cette approche se situe entre l'approche « cycle en V » et l'approche strictement agile comme Scrum. Ce mode d'organisation de projets combine les avantages du cycle en V et de la méthodologie agile :

- La structuration de la méthodologie V est conservée : les développements des prochains cycles sont définis voire spécifiés dans le détail pour certains ;
- La rapidité de la méthodologie agile y est intégrée : nous sommes sur des cycles variant de 3 à 6 semaines (similaires à ceux pratiqués en Scrum) ;
- Les lots sont plus petits et les validations plus fréquentes ;
- Le projet se construit plus rapidement sur des bases validées fréquemment ;
- Implication permanente du client.

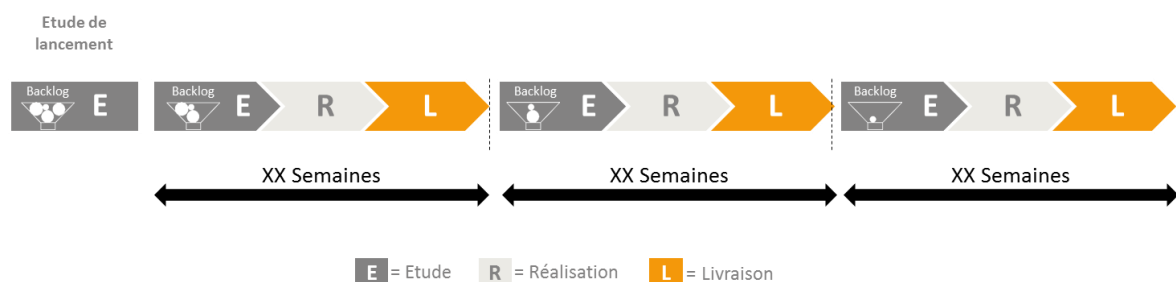


Figure 30 - Méthodologie projet informatique X2i : le cycle itératif « agile ».

Sur ce projet chaque itération débute par une séance de travail avec le client lui présentant l'état du projet, l'accord sur les différents points à adapter puis termine par les décisions et directives pour le prochain cycle. La cartographie du processus a été révisée 6 fois avant sa transposition en BPMN 2.0.

L'aboutissement de la création de ce prototype a été divisé en trois familles de cycles chacune avec des livrables intermédiaires.

1. **Descriptive et graphique** : ayant pour objectif la reformulation du processus P1 dont le livrable est sa modélisation en utilisant la notation BPMN 2.0.
2. **Formalisation** : dont le livrable est l'intégration des modèles BPMN 2.0 dans le moteur de workflow Activiti et définition des modèles dans Alfresco.
3. **Exécution** : dont le livrable est l'intégration et instanciation du workflow Activiti dans Alfresco.

Le livrable de cette phase est le produit fini ainsi que sa documentation.

IV.2.4.3 Phase de livraison finale

La phase de livraison sur ce projet a abouti avec le procès-verbal de recette des livrables suivants.

- Création d'un environnement démontrable du prototype sous la forme d'une machine virtuelle autonome au format *Virtual Box*.
- Soutenance de la méthodologie appliquée à l'équipe en présentant la :
 - Méthode de formalisation d'un processus en BPMN 2.0 ;
 - Méthode d'intégration de ce processus dans le moteur de workflow Activiti ;
 - Méthode d'intégration de ce processus dans Alfresco ;
 - Démonstration.
- Rédaction d'une documentation de synthèse en anglais sur la méthodologie appliquée.
- Fourniture des codes sources :
 - *Workspace* du projet Activiti *Designer* contenant la configuration Alfresco et l'archive .bar du processus NS-P1 ;
 - Scripts *Apache Ant* des tâches automatisées du déploiement de ces fichiers sur une instance d'Alfresco.
- Fourniture d'un jeu de données afin de pouvoir tester le processus.
- Archives de tous les outils utilisés par le projet.
- Journal de développement contenant la liste des activités et de recherches bibliographiques.
- Export au format XMI de la modélisation faite sur l'outil de modélisation UML *Enterprise Architect*.

IV.3 La plate-forme Activiti

Cette section présente les composants de la plate-forme Activiti utilisés pour spécialiser le processus BPMN 2.0 en processus exécutable sur le moteur de workflow Activiti puis sur Alfresco.



Activiti est une plate-forme logicielle *BPM* et de workflow légère qui vise les responsables métiers, les analystes métier, les développeurs et les administrateurs système. Elle fournit un moteur d'exécution BPMN 2.0 fiable et rapide utilisant la technologie *Java*. La plate-forme Activiti est légère fondée sur des normes ouvertes et est conçue pour s'intégrer dans des logiciels utilisant le framework *Spring* comme Alfresco mais propose aussi une API REST. La dernière version en date est la 5.17.0 publiée le 18 Décembre 2014.

Ma démarche était d'étudier différents facteurs sur cette plate-forme pour justifier son emploi dans le projet *Acoustic*. Historiquement sur Alfresco les processus métier sont définis en utilisant un formalisme à part appelé JBoss BPM (ou jBPM). La raison principale pour le choix de la plate-forme Activiti est la volonté de ne pas multiplier les différentes technologies et de rester sur un standard mondial et pas une spécificité technologique isolée. Le standard BPMN 2.0 est utilisé par Activiti mais aussi par d'autres moteurs de workflow compatibles comme *Bonitasoft* par exemple. Un grand nombre d'éditeurs graphiques de modélisation UML propose la création de diagrammes BPMN 2.0 et de les exporter en XML pour exécution.

Les différents facteurs étudiés pour valider le choix technologique d'Activiti sont énumérés ci-après :

- facilité de prise en main
- niveau et qualité de la documentation (11)
- activité de la communauté
- réactivité de la communauté
- outils d'édition de de gestion
- intégration dans des systèmes tiers
- possibilité de faire des tests unitaires
- dysfonctionnements
- compatibilité BPMN 2.0

Par exemple j'ai été agréablement surpris par la réactivité concernant la remontée d'un dysfonctionnement sur leur éditeur basé sur l'EDI Java *Eclipse*. J'ai remonté le « bug » et une suggestion de correctif sur leur forum qui a été intégré dans la version suivante (<http://forums.activiti.org/content/execution-listener-only-task-listener-allowed-wizard>).

Activiti ne propose pas seulement un moteur de workflow mais plusieurs outils complémentaires pour gérer des processus BPMN 2.0 de la définition de processus à son exécution puis à son suivi. Une vue d'ensemble de cette pile d'outils est représentée dans la Figure 31 :

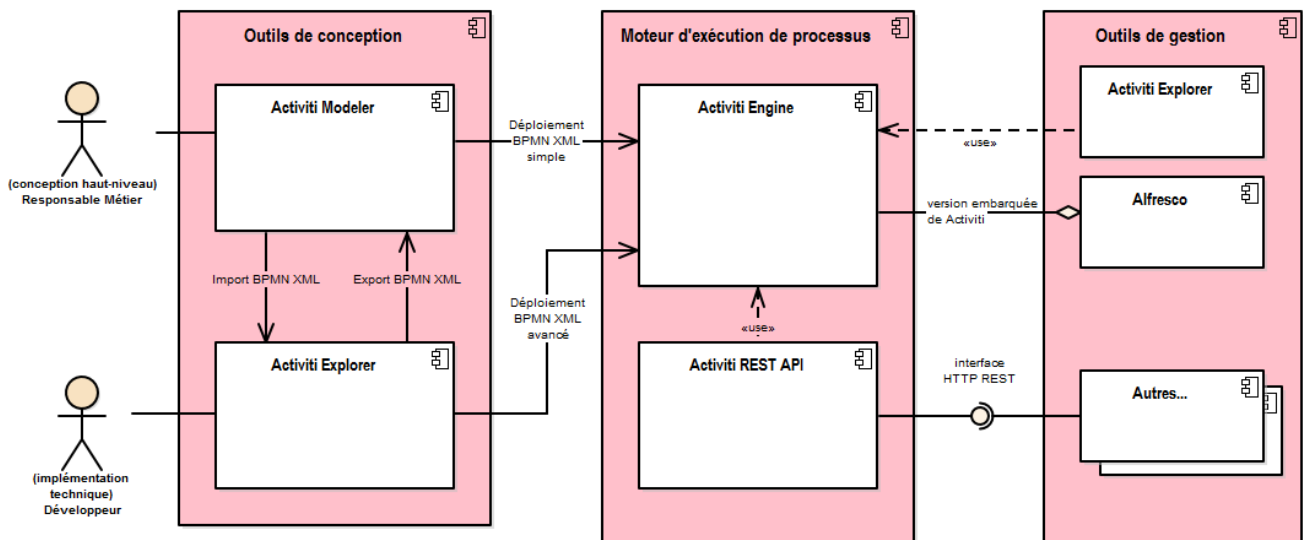


Figure 31 - Vue d'ensembles des outils Activiti³²

Le moteur d'exécution Activiti peut être déployé seul mais il est nativement embarqué avec Alfresco³³.

En résumé et avec le support de la figure précédente : afin de pouvoir exécuter un processus nous avons besoin d'un part de concevoir et de modéliser le processus, d'autre part d'instancier le processus et de l'exécuter puis finalement de pouvoir le gérer et suivre son avancement.

Les rôles de chaque outil dans la définition de processus pour le moteur d'exécution Activiti seront décrits dans les chapitres « Présentation des outils » des parties « Modélisation » et « Instanciation ».

En revanche, une nouvelle problématique se pose : si nous voulons informatiser un processus nous devons utiliser un moteur de workflow et cela implique une adaptation de la syntaxe BPMN 2.0 pour ce moteur. C'est le premier **rempart technologique** sur l'instanciation de processus sur des solutions technologiques hétérogènes. Ce rempart est le choix du moteur d'exécution. Pour le projet *Acoustic* nous avons opté pour Activiti, nous allons voir plus loin dans ce mémoire que pour le projet *Descours & Cabaud*³⁴ nous avons construit un moteur d'exécution sur mesure. Le point important à retenir est que l'avantage d'utiliser le formalisme BPMN 2.0 est qu'un processus est décrit dans un premier temps sans devoir prendre en considération la solution technique permettant son exécution. Dès que la solution technique est identifiée nous devons adapter³⁵ la définition de processus BPMN 2.0 pour le moteur d'exécution.

³² Au centre le moteur de workflow, à gauche les outils de modélisation et à droite l'outil de gestion.

³³ A partir de sa version 4, Alfresco embarque le moteur Activiti nativement.

³⁴ Cf. chapitre V.1 « Cas 2 : Descours & Cabaud et les Circuits de validation »

³⁵ L'adaptation d'un processus métier pour son exécution sera présentée dans la section « Instanciation de processus » page 41.

Conceptuellement nous avons besoin de suivre différentes étapes.

1. Opter pour un formalisme de définition de processus, nous en connaissons les avantages : ici BPMN 2.0.
2. Rédiger les premières versions d'une définition de processus sur laquelle toutes les parties prenantes du projet sont d'accord. La valider tout en gardant à l'esprit que le travail de définition de processus est itératif sous la forme de cycle d'amélioration continu. Nous allons voir plus loin que la discipline *BPM* définit ce cycle comme le « Cycle BPM ».
3. Identifier les couches logiciels cibles qui exécuteront ce processus : ici Activiti.
4. Adapter³⁵ la définition BPMN 2.0 en fonction du moteur d'exécution si le moteur est conçu pour être compatible avec le XML sous-jacent (ou XML-BPMN 2.0).

IV.3.1 Activiti Engine

En informatique afin de pouvoir instancier et exécuter un processus, nous avons vu précédemment qu'il est nécessaire de le rendre exécutable par un moteur de workflow.

Le « Moteur Activiti » (ou *Activiti Engine*) est le composant cœur de Activiti. C'est lui qui exécute les définitions de processus en BPMN 2.0 déployées à partir de sources XML. Puis est capable d'instancier un processus et crée chaque tâche du processus tout en préservant un historique des tâches et instances des processus exécutés. Ce moteur, codé en Java, est intégrable directement dans des applications compatible *Spring*.

IV.3.2 Activiti Designer

L'outil Activiti Designer de la plate-forme Activiti est orienté développement et intégration du processus dans le moteur de workflow Activiti et Alfresco. Ici le processus est rendu exécutable, pour ce moteur, et contient toutes les variables d'exécution, les définitions de formulaires pour interaction, les services et scripts contenant des instructions logiques et finalement la possibilité de définir des tests unitaires afin d'assurer un niveau de qualité interne logicielle. Cet outil est intégré sous la forme d'extension à l'EDI Eclipse très répandu chez les équipes de développement.

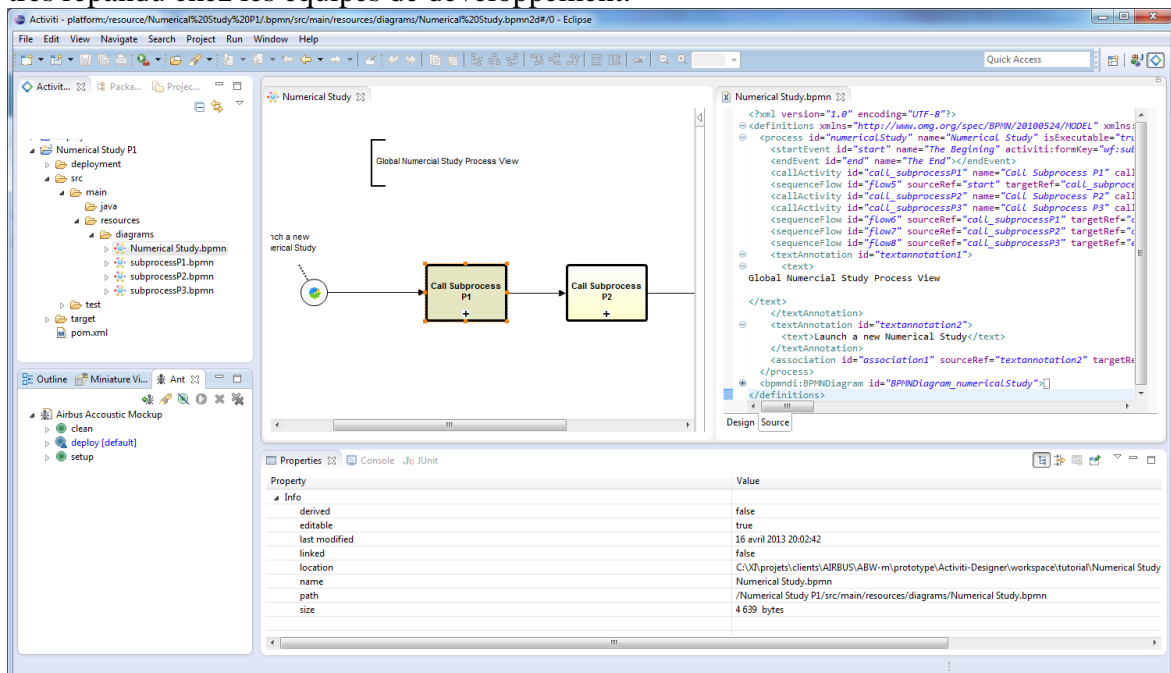


Figure 32 - Ecran de l'espace de travail du plug-in Eclipse : Activiti Designer.

Le moteur Activiti permet la création de tests unitaires permettant d'améliorer la qualité interne logicielle avec la pratique du TDD³⁶, j'ai l'habitude de mettre cette pratique en œuvre sur mes projets de commerce électronique mais, malheureusement ça n'a pas été inclus dans le cadre du projet *Acoustic* puisque l'objectif était la réalisation d'un prototype. Par contre elle a été mise en œuvre sur le projet *Descours & Cabaud*. Un autre atout d'un environnement de développement Java est la possibilité d'automatiser des tâches de développement récurrentes avec l'outil *Apache Ant*³⁷. Notamment le déploiement des définitions de processus et des configurations des modèles de données Alfresco et configurations IHM *Share Forms* et le redémarrage des *Servlets* du serveur d'applications *Tomcat*.

Le plug-in Eclipse Activiti Designer, par-contre, est malheureusement doté de plusieurs dysfonctionnements. Si vous en avez connaissance et des moyens pour les contourner il est possible de travailler dans un environnement de développement quelquefois frustrant mais très correct qui est tout de même mieux que l'édition d'un fichier XML directement. Tous les dysfonctionnements identifiés et leur contournements ont été réunis dans un document de journal de développement afin qu'il puisse servir à d'autres personnes pouvant travailler sur le projet *Acoustic* dans un souhait de partage de connaissances.

IV.3.3 Activiti Explorer

L'outil Activiti Explorer de la plate-forme Activiti est une application web de gestion des processus déployés sur le moteur Activiti. Il est possible par exemple d'instancier un processus, suivre l'avancement des tâches, assigner des tâches, consulter une liste de tâches assignées à un utilisateur et consulter des rapports.

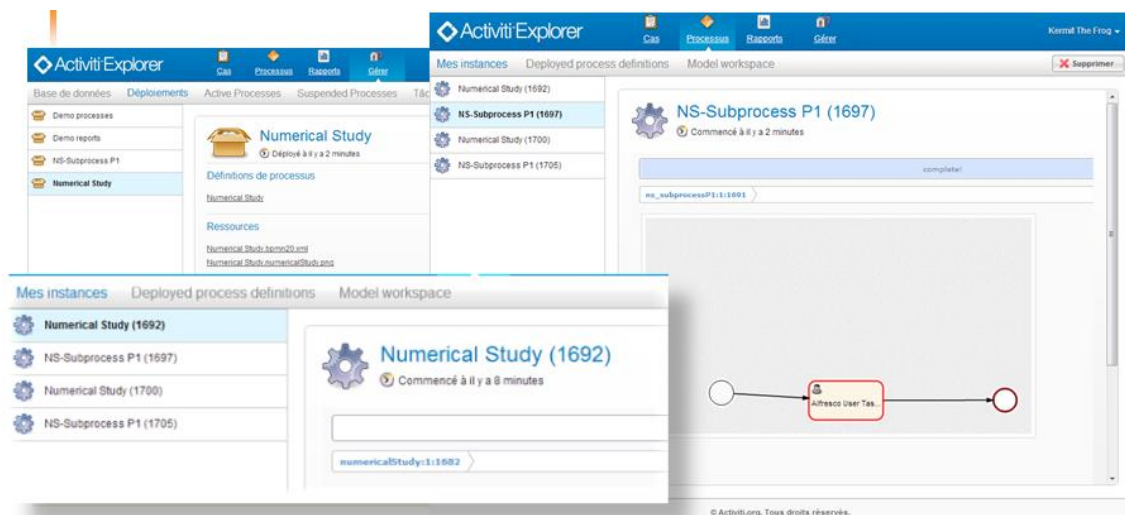


Figure 33 - Ecrans de l'application web Activiti Explorer

Cet outil est primordial pour simuler et valider un processus avant son intégration dans la solution finale. Pourquoi ce type d'outil est-il primordial ?

³⁶ *Test Driven Development* ou développement piloté par les tests (par exemple JUnit pour Java et PHPUnit pour PHP).

³⁷ Outil applicatif de la fondation Apache qui vise à automatiser des opérations répétitive du développement de logiciels.

Un processus est une séquence d'activités, si ce processus est informatisé certaines activités ont besoin d'interactions humaines se basant sur un interfaçage homme machine, ou IHM. Dans ce cas il faut décrire quels sont les champs de ce formulaire et à qui il sera assigné, il faut aussi définir quels variables seront portés par le processus afin de maintenir un état sur les informations véhiculées.

L'outil Activiti Explorer permet justement d'instancier un processus puis de simuler les données qui seront véhiculées, les formulaires affichés et les rôles à qui les différentes tâches seront assignées. Et ce avant l'étape exécution sur la solution logicielle finale.

Une activité dans un processus métier informatisé est donc une tâche affectée à une personne humaine auquel est associé un formulaire qui utilise et crée des variables de processus.

L'exemple de la Figure 34 sur un processus de demande de congés démontre ce point. Les formulaires d'interactions humaines sont encadrés en rouge et associés avec un événement de départ (le formulaire qui configure l'instance du processus qui sera exécuté) et le *User Task* « Traite demande de congés ».

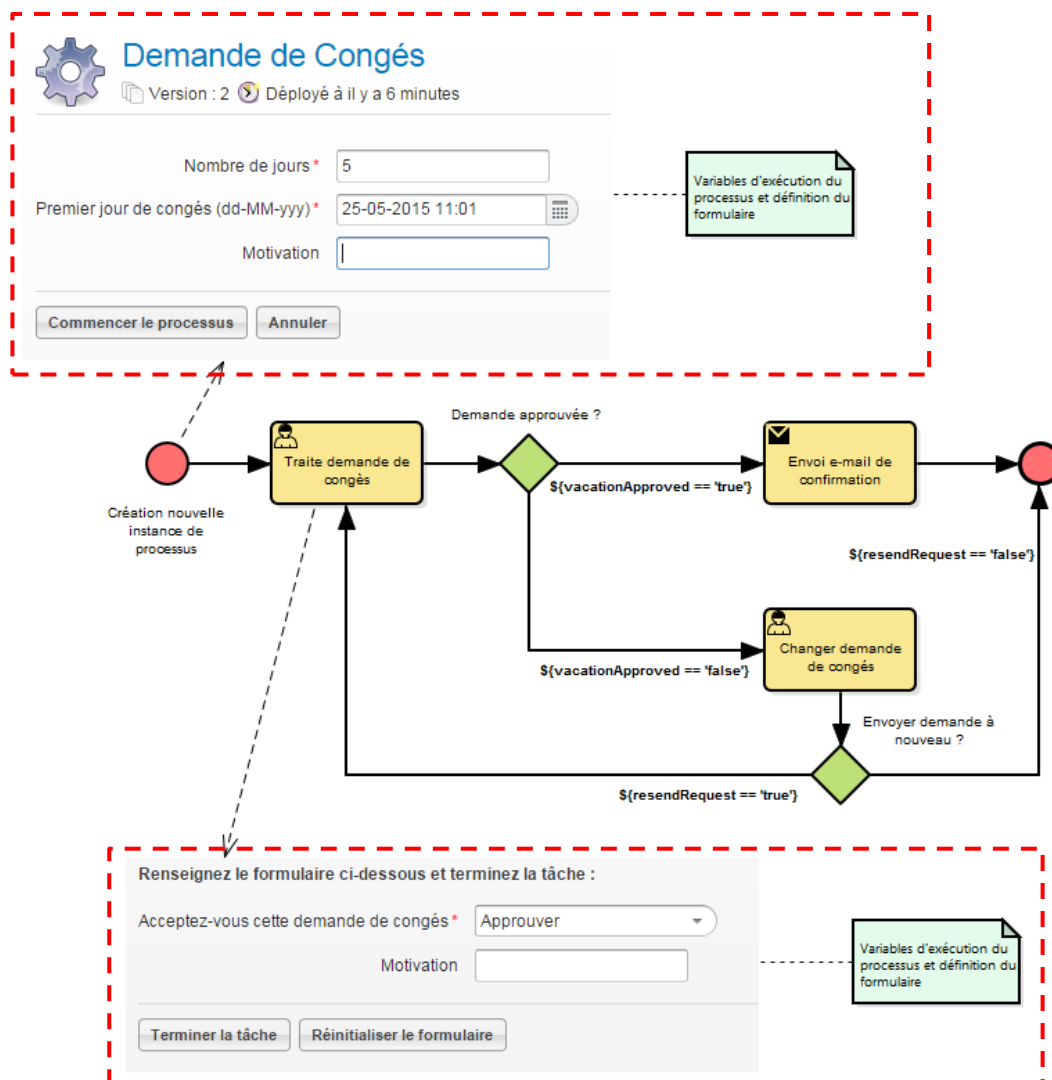


Figure 34 - Processus BPMN 2.0 montrant des IHM de tâches d'interactions humaines extraites de l'outil Activiti Explorer.

Le premier formulaire permet de démarrer un processus de demande de congés et le second de le valider ou de le refuser. Activiti étend la spécification BPMN 2.0 par ses propres points d'extensions tout comme d'autres moteurs de workflow comme *Bonitasoft* par exemple.

Ici les points d'extensions utilisés permettent :

- d'assigner un ou plusieurs rôles aux différents *User Tasks* ;
- d'associer un formulaire et la définition de ses champs avec les éléments *Start Event* et *User Tasks*;
- d'écrire le script de la logique des branchements ;
- de définir le script d'envoi de courrier électronique ;
- de définir les variables d'exécution et les variables de processus.

IV.4 Méthodologie

Un processus humain peut être modélisé et des bénéfices peuvent en ressortir. Le premier rempart technologique est le choix de son implémentation : manuelle, semi manuelle avec l'aide de logiciels ou complètement informatisée. Dans le choix de l'implémentation il faut entendre : « par quel moyens le processus sera exécuté ? ». Un moteur de workflow est un dispositif de centralisation permettant l'exécution des instances de processus (ou workflow). Il permet d'automatiser les branchements, d'assigner des tâches à des acteurs humains et d'exécuter des instructions programmatiques. Mais aussi offre la possibilité de suivre l'avancement des tâches, de lister les tâches sur lesquelles une personne (un rôle) doit faire un traitement, de générer des rapports d'exécution, etc.

Pour le projet *Acoustic*, le choix est d'utiliser Alfresco qui propose un moteur de workflow extensible avec le moteur de workflow Activiti. Historiquement sur Alfresco un autre moteur de workflow était utilisé, jBPM, et Activiti est son successeur³⁸.

En théorie tout outil capable de produire un XML conforme à la spécification BPMN peut être utilisé pour modéliser des processus métier et tout moteur de workflow conforme à la spécification BPMN 2.0 peut être utilisé pour exécuter un processus. Le XML résultant de la définition BPMN 2.0 devrait en conséquence pouvoir s'exécuter sur un moteur de workflow compatible. La difficulté réside dans la spécialisation de la syntaxe BPMN 2.0 pour que sa définition puisse d'exécuter telle qu'elle a été conçue pendant la phase de modélisation. Le niveau et la complexité de cette spécialisation dépendent du moteur de workflow retenue et de sa qualité. Et le choix du moteur de workflow impose des contraintes et des limitations qui peuvent impacter la conception initiale.

Ce chapitre présentera l'implémentation du processus NS sur le moteur de workflow Activiti en spécialisant sa définition XML-BPMN 2.0 pour qu'elle puisse s'interfacer avec lui. C'est l'adaptation d'une définition de processus dans les étapes de transmutation de processus.

³⁸ Tom Baeyens, le créateur de jBPM, et certains membres de son équipe ont conçu le projet open source Activiti avec l'appui d'Alfresco pour créer le moteur de workflow compatible BPMN Activiti.

Ensuite le processus sera rendu exécutable sur le moteur de workflow Activiti. Nous allons voir que cette spécialisation de processus entraîne des contraintes dans la conception même du processus qui devra en conséquence être adapté.

IV.4.1 Intégration dans Activiti Engine

Une de mes premières tâches d'analyse était de comprendre comment Activiti fonctionne (12) en relation avec la spécification BPMN 2.0 de l'OMG. Elle contient plus de 500 pages et le moteur Activiti se base sur un sous-ensemble de catégories suffisant pour exécuter des processus métiers couvrant le niveau **basique**, **cœur** et quelques éléments **avancés**³⁹.

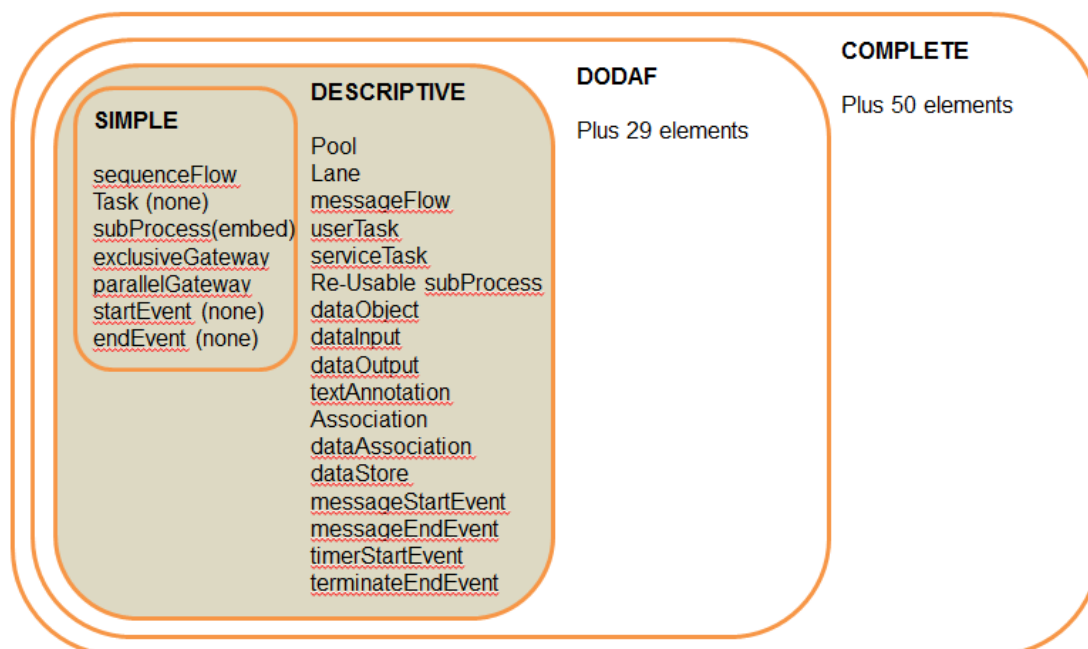


Figure 35 - Mise en avant des catégories (ou niveaux) BPMN 2.0 utilisées par Activiti.

Par contre pendant cette phase d'études j'ai constaté plusieurs dysfonctionnements de la version du moteur Activiti utilisée et j'ai dû trouver des solutions alternatives. Par exemple, même si les *Pools* et les *Lanes* font partie des éléments BPMN supportées par Activiti, elles ne fonctionnent pas correctement et doivent être abandonnées. La solution appliquée est de préfixer chaque *User Task* par une abréviation du rôle qui en est responsable. Cela permet de supprimer syntaxiquement les *Lanes* sans que la représentation graphique perde son sens initial.

IV.4.1.1 Assignation de tâches

Un *User Task* représente est une série d'étapes qu'un humain fait avec une IHM (souvent un formulaire). Les rôles responsables du déroulement d'un *User Task* doivent y être explicitement assignés. Les acteurs issus de la modélisation représentent chacun un rôle et dans Alfresco une personne s'authentifie sur la plateforme avec un compte utilisateur et

³⁹ Sur le projet *Acoustic* aucun élément du niveau **avancé** n'a été utilisé.

peut être associé avec un groupe ayant des droits particuliers sur les documents de la plateforme. Activiti assigne soit des utilisateurs soit des groupes d'utilisateurs à des tâches.

Le diagramme de la Figure 36 va tenter de clarifier cette projection des acteurs du processus NS-P1 identifiés pendant la phase de modélisation sur Activiti et Alfresco.

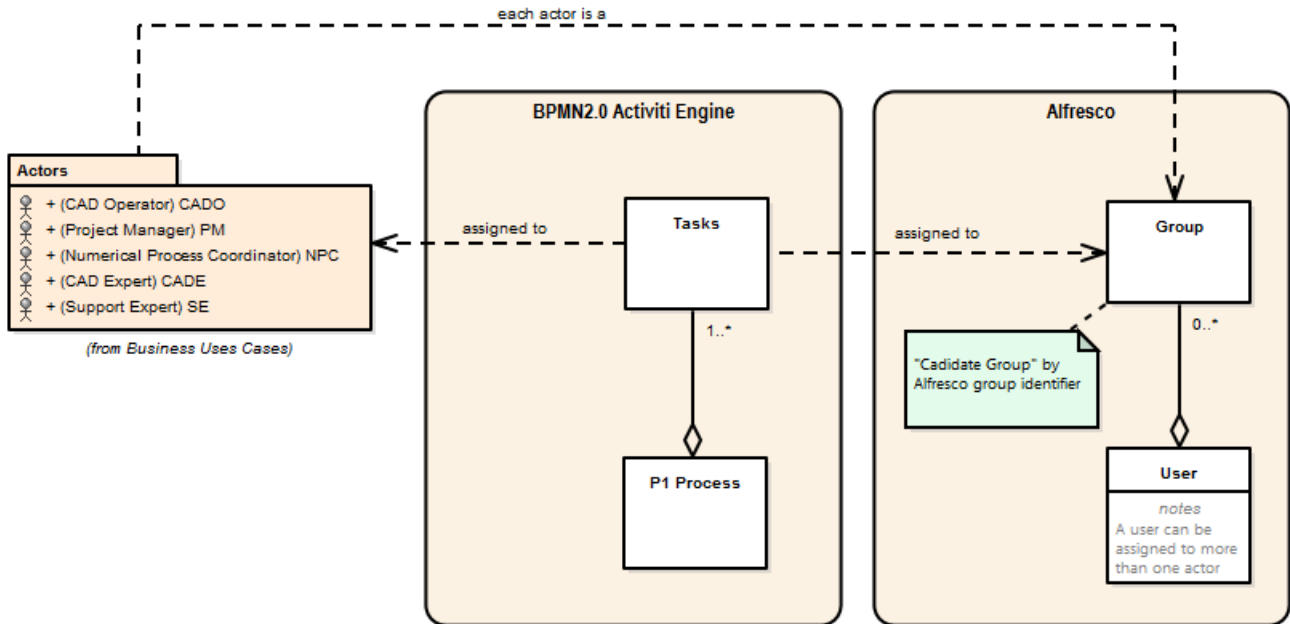


Figure 36 - Projection des Acteurs sur Activiti et Alfresco.

Techniquement l'outil Activiti Designer ajoute avec le *namespace* de son extension de définition XML-BPMN 2.0 un attribut d'élément XML : **activiti:candidateGroups** avec comme valeur l'identifiant Alfresco du groupe associé avec l'acteur.

```
<userTask id="pm_ongP1KOM" name="PM: Organize M1 KOM" activiti:candidateGroups="GROUP_ns-pm" >
```

Chacun des *User Task* de la définition de processus NS doit être modifié de la même manière. Cette modification n'affecte que le XML sous-jacent et la représentation graphique du processus reste inchangée. Sans l'aide de l'outil Activiti Designer ce serait une opération très lourde, la lecture de documents XML étant très indigeste.

Cette première spécialisation est importante puisqu'elle associe, de manière informatique, un compte utilisateur avec une activité du processus. Seul un utilisateur de la solution logicielle cible ayant les autorisations suffisantes pourra intervenir sur cette activité pendant l'exécution de cette activité. Activiti propose différentes associations possibles :

- **Assignment directe (assignee)** : permet l'assignation directe d'une tâche à un utilisateur de la plate-forme cible.
- **Utilisateur éventuel (candidateUsers)** : définition d'une liste d'utilisateurs qui auront le potentiel d'être assigné à une tâche, le premier qui accepte l'assignation y est associé.
- **Groupe d'utilisateurs éventuels (candidateGroups)** : comme précédemment avec la particularité de lister des groupes d'utilisateurs.

Cette extension d'assignation n'est pas un standard BPMN 2.0 et ne sera pas représenté graphiquement d'autant plus qu'il est possible de déléguer cette assignation à une classe Java pour injecter un comportement spécifique. La modélisation BPMN 2.0 ne permet pas

de représenter ces cas spécifiques mais ils peuvent figurer dans les scénarios des cas d'utilisation associées ou plus haut dans la spécification. Une contrainte de l'implémentation dans un moteur de workflow est la diffusion de comportement métier et une documentation à jour est nécessaire sinon nous injectons des risques de maintenance.

Pour le projet *Acoustic* nous avons opté pour la souplesse de l'utilisation des « Groupes d'utilisateurs éventuels » permettant d'associer une tâche à des groupes d'utilisateurs Alfresco. Pour des projets ayant une solution différente, l'utilisation des API's Activiti rendent possible cette relation utilisateur / interprète.

IV.4.1.2 Formulaires d'interaction

Les formulaires dans un workflow permettent à un humain de manipuler les données (ou attributs) d'une tâche permettant de passer à l'activité suivante dans l'ordre d'exécution du processus. Typiquement ces manipulations sont de l'ordre de :

- l'association de documents ;
- la définition d'une date d'échéance ;
- la saisie de commentaires ;
- la validation ou le refus d'une demande de validation ou confirmation ;
- etc.

Commencer un processus sous la forme d'une nouvelle instance et participer à ses tâches permettent à des personnes de participer activement à un processus. Afin qu'un logiciel puisse communiquer avec ces personnes il a besoin que des formulaires puissent être affichés avec une technologie IHM.

A chaque tâche d'interaction humaine il faut associer un formulaire. Le moteur Activiti propose deux stratégies pour l'association de formulaires avec des tâches.

1. L'utilisation d'une **définition de formulaire embarqué**.
2. **Délégation à l'applicatif client** de cette définition.

La **définition de formulaire embarqué** d'Activiti a plusieurs avantages puisqu'elle positionne la responsabilité de la gestion des champs du formulaire au moteur. Par API il est possible de récupérer cette définition de formulaire et l'exploiter sur le client. L'élément XML `<activiti:formProperty>` est utilisé comme point d'extension dans la définition XML des *User Task* et du *Start Event* du processus.

La difficulté côté client est de devoir créer un interpréteur de la définition de formulaire faite avec Activiti.

Alfresco propose une gestion technologique de formulaires avancés avec l'application Share⁴⁰. D'un point de vue de l'architecture, le moteur de formulaire Alfresco *Share Forms* est la combinaison de plusieurs composants de gestion de formulaires. Ce moteur est constitué par les quatre composants suivants.

⁴⁰ Alfresco Share est une application web cliente ajoutant une interface utilisateur collaborative, extensible et ergonomique sur le « Repository Alfresco » ou plus précisément l'Alfresco Content Application Server. Il remplace l'ancien client « Alfresco Explorer » et est implémenté sur la technologie Spring Surf.

- **Form Service** : responsable de l’affichage des données que le formulaire doit gérer et de persister les modifications faites suite à la soumission d’un formulaire. Il récupère les données et les associe à des *Form Components* pour les représenter visuellement.
- **Form Component** : responsable de l’affichage des types de composants (ou champs) d’un formulaire : champ texte, champ date, champ téléchargement de fichier, champ informatif, champ de navigation dans une arborescence de fichiers etc. Sémantiquement un formulaire est un ensemble de champs typés ayant un libellé, une validation de contenu et une description ou notice d’information explicative de ce champ pour l’utilisateur. Ces champs peuvent être regroupés et agencés en sections logiques. Chaque champ représente une ou un ensemble de données. A chaque composant est associé un gabarit par défaut au format *Freemarker*.
- **Form Configuration** : responsable de la configuration du formulaire et de l’agencement des différents champs (ou Form Components).
- **Form Runtime** : responsable du comportement dynamique sur la page web du formulaire.

Cette architecture est suffisamment découplée et a l’avantage d’être souple. Par contre cette architecture est complexe et la prise en main est longue. Pour les besoins du projet *Acoustic* nous avons opté pour la gestion des formulaires avec *Share Forms* parce que ce moteur est déjà intégré dans Alfresco.

La gestion de formulaires par **délégation à l’applicatif client** revient simplement au positionnement d’un identifiant unique de formulaire sur un point d’extension XML dans la définition de processus. Le moteur Activiti retourne cet identifiant et délègue la définition du formulaire au client. Cette solution, à mon sens, est mieux adaptée aux systèmes hétérogènes et laisse au client la responsabilité de la gestion de ses formulaires. L’attribut XML **activiti:formKey=""** est utilisée comme point d’extension de *namespace* dans la définition de processus. Dans le cadre de l’intégration dans Alfresco le moteur Activiti retourne l’identifiant de formulaire à Alfresco qui à son tour le fournit au moteur *Share Forms* pour l’affichage du formulaire.

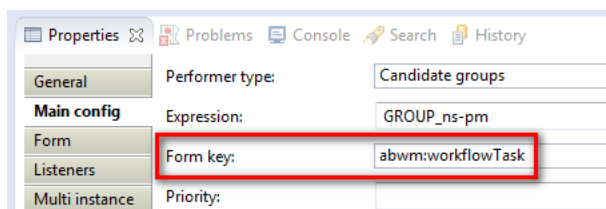


Figure 37 - Configuration de gestion de formulaires par délégation dans Activiti Designer.

Il est important à ce stade de concevoir des maquettes des formulaires et de les confronter aux modèles de données réalisées pendant la phase de modélisation. Idéalement une simulation est nécessaire afin de pouvoir manipuler le processus. L’intégration de formulaires par délégation peut être longue et complexe, par exemple avec Alfresco Share, et le processus n’est pas encore complètement exécutable. Les règles de validation de champs sont définies ici.

Pendant la réalisation du projet *Acoustic*, afin de préparer cette séance de travail, j'ai utilisé Alfresco Explorer pour simuler un processus indépendamment d'Alfresco. Dans *Activiti Designer* j'ai rapidement préparé provisoirement les formulaires pour chaque tâche avec la stratégie de définition de formulaires embarquées et fait des scripts rapides pour parcourir les branchements en fonction de l'issue des formulaires. Je suis convaincu des avantages de la simulation d'un modèle et la présentation de ce concept appliqué au processus NS à une équipe spécialisée dans les études numériques, a été bien accueillie.

Au cours de la séance de travail nous avons simulé le processus NS et découvert des nouvelles données qui ont abouti à une évolution du modèle de données du processus.

L'issue de formulaires représente la décision prise pendant la soumission du formulaire. Les données saisies dans le formulaire sont ensuite transmises pour effectuer des tests logiques au niveau des branchements. Par exemple : je valide ou refuse un document et saisis la raison, je termine une tâche, etc.

Avec le support du modèle de données, les maquettes de formulaires et la définition des issues des formulaires : il est possible d'implémenter ces formulaires dans la solution cible, ici Alfresco Share.

La Figure 38 montre une vue IHM d'un formulaire *Share Forms* de *User Task* avec issue.

Edit Task: NPC: Launch Numerical Study Release to Pool

* Required Fields

Info

Numerical Study Name: The Study

Owner: Administrator Priority: Medium Due: (None)

Identifier: 47882

Assignee

P1 Role:

Numerical Process Coordinator

Progress

Status: *

Not Yet Started

Response

M1 Report:

Comment:

Approve Reject

Assignment du User Task à un rôle

Issue du formulaire

Figure 38 - Formulaire *Share Form* d'un *User Task* du processus NS-P1.

IV.4.1.3 Injection de la logique

Au démarrage d'une nouvelle instance d'un processus des variables sont créées et seront accessibles tout au long de la vie du processus. Chaque formulaire pourra enrichir cette liste de variables et par la suite lire et modifier les valeurs. Ce sont les **variables de**

processus. Pendant l'exécution d'une activité il est possible d'utiliser des variables locales appelées **task variables**. Cet ensemble de variables forme le **modèle de données d'exécution d'un processus Activiti**⁴¹. Elles sont persistées avec l'état de l'instance du processus et forment des paires *clef-valeur*.

Chaque nouvelle instance de processus a le potentiel de suivre une séquence d'activités différente que la précédente. Les valeurs des formulaires peuvent ne pas être les mêmes, ou certaines valeurs définies au démarrage du processus peuvent varier. Au-delà d'un document ayant besoin d'une validation, un certain nombre de données complémentaires et requises pour le bon fonctionnement des branchements est nécessaire. Un moteur de workflow doit pouvoir proposer la fonctionnalité de collection de variables accessible en lecture/écriture pendant l'exécution.

La Figure 39 résume la relation entre les éléments d'un processus, les variables et les formulaires sous la forme de diagramme de domaine.

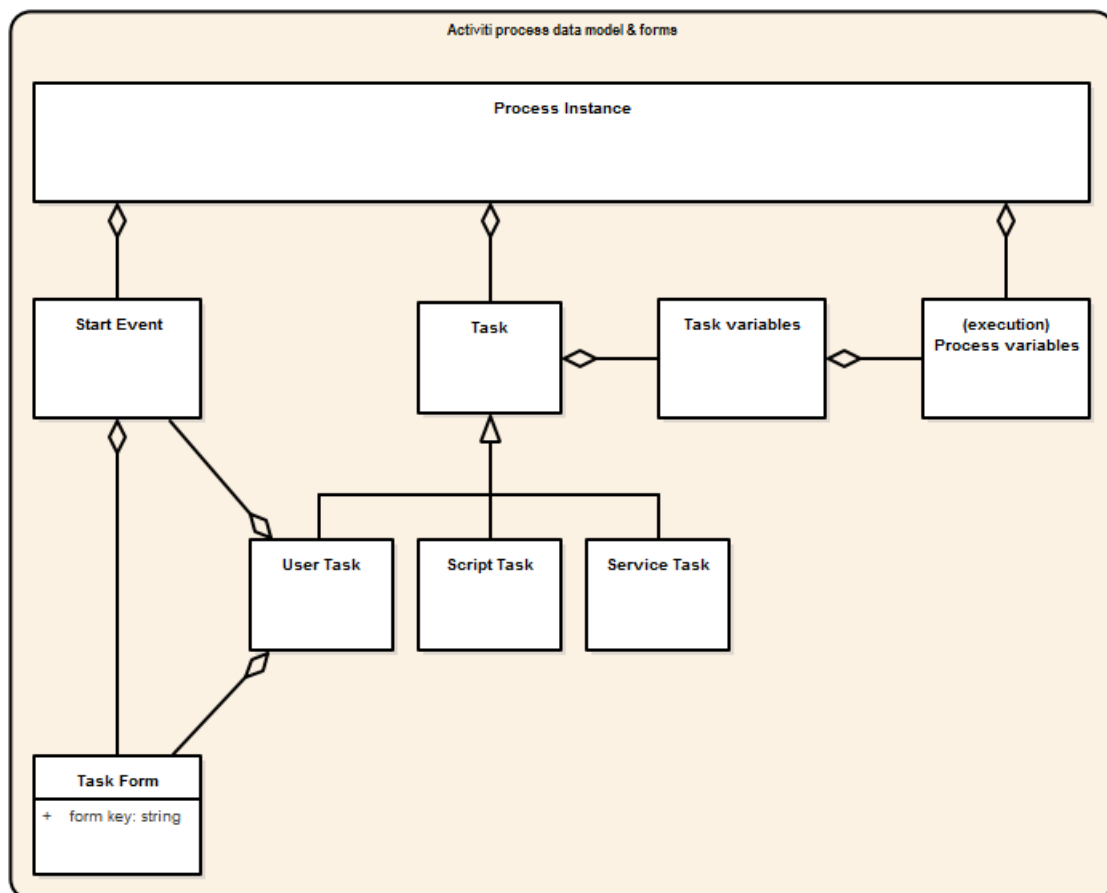


Figure 39 - Modèle de données d'un processus Activiti.

Nous pouvons observer qu'un *Start Event* et un *User Task* ont des formulaires et qu'une instance de processus a des Process Variables qui sont accessibles par chaque *Task* via l'objet *Execution* dans le moteur Activiti.

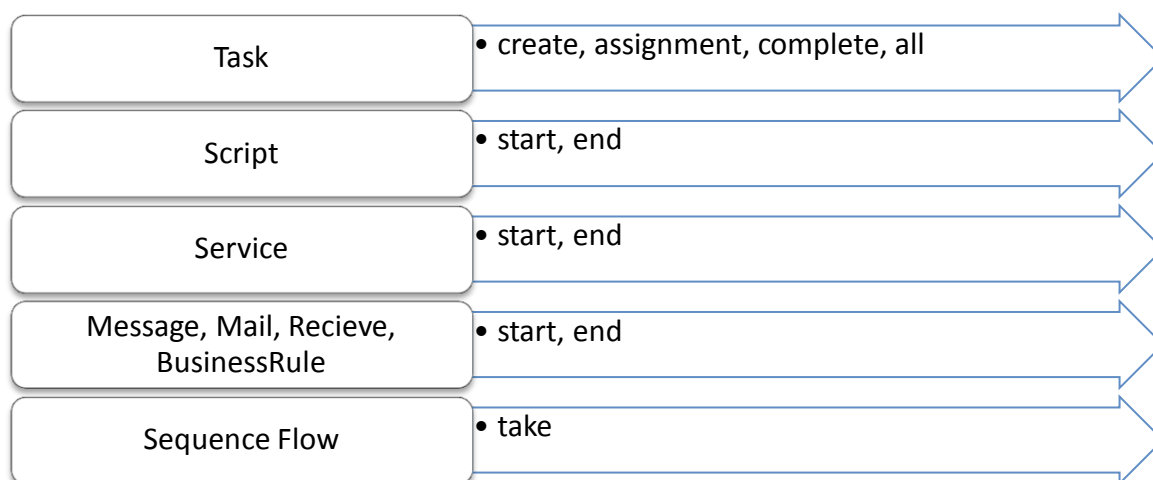
⁴¹ « *Activiti process data model* » en anglais.

La solution Activiti propose des API pour manipuler ces données à partir de client tiers, mais dans le cadre d'une intégration Alfresco, ces données sont persistées par les services de persistance d'Alfresco.

Comment injecter de la logique d'exécution dans un processus ? Le moteur Activiti en prévoit plusieurs : système d'observateurs, *Script Task*, *Service Task* et scripts de branchements.

Etant donné que le XML de définition de processus est extensible, le moteur Activiti ajoute des nouveaux points d'extensions spécifiques permettant l'ajout d'**observateurs**⁴² qui se déclenchent en fonction d'une transition d'état sur l'exécution d'un élément graphique. Ces observateurs se positionnent directement sur des **éléments** (*Start Event*, *User Task*, *Service Task* et *Script Task*) du processus et des **associations** (ou *Sequence Flows*) entre eux.

Voici la liste des transitions sur lesquelles il est possible de positionner un observateur qui exécutera une logique automatique sous forme de programme.



Cette logique a pour objectif de manipuler les variables d'exécution du processus mais pas l'injection d'un nouveau comportement ou branchement ceci doit être fait au niveau de la modélisation BPMN (c'est son rôle). Il faut garder à l'esprit que n'importe quel élément d'une représentation graphique BPMN 2.0 est repositionnable, réutilisable et remplaçable, si une logique comportementale est injectée par l'utilisation de ce système d'observateurs, il sera emporté avec l'élément et le processus risque de ne plus pouvoir s'exécuter ou avoir un comportement imprévu. C'est essentiellement pour cette raison que la pratique d'implémentation d'un processus sur un moteur de workflow adapte la définition de processus initial.

Ce système d'observateurs est capable de déclencher l'exécution de script ou déléguer l'exécution à des classes Java. Activiti prévoit un système de langage d'expression permettant de simplifier la syntaxe des scripts.

⁴² « Listeners » en anglais.

Une problématique majeure émerge de l'implémentation d'un processus, étant donné qu'il est fortement couplé avec un moteur de workflow particulier, la portabilité d'un processus est réduite ou en tout cas polluée par les instructions XML spécifiques à un moteur.

La spécification BPMN 2.0 prévoit des éléments dont l'exécution est automatique utilisée par Activiti, par exemple les *Script Task* et les *Service Task*, ou encore l'implémentation de la logique de branchement. Des points d'extensions sont encore utilisés pour les implémenter.

Afin de pouvoir implémenter un processus tous les **branchements exclusifs** (ou *Exclusive Gateways*) doivent être programmés. Ce sont les *Sequence Flows* sortants qui portent la logique et non pas le branchement lui-même dans Activiti. La logique est implémentée sous la forme de prédicat sur chaque *Sequence Flow* suivant le branchement exclusif. Pour qu'un *Sequence Flow* puisse être parcourue, le prédicat doit répondre par vrai.

La Figure 40 et la Figure 41 montrent des exemples de prédicats de choix de branchements automatiques avec l'utilisation du langage d'expression d'Activiti dans le champ Condition.

Prédicat pour un branchement de validation.

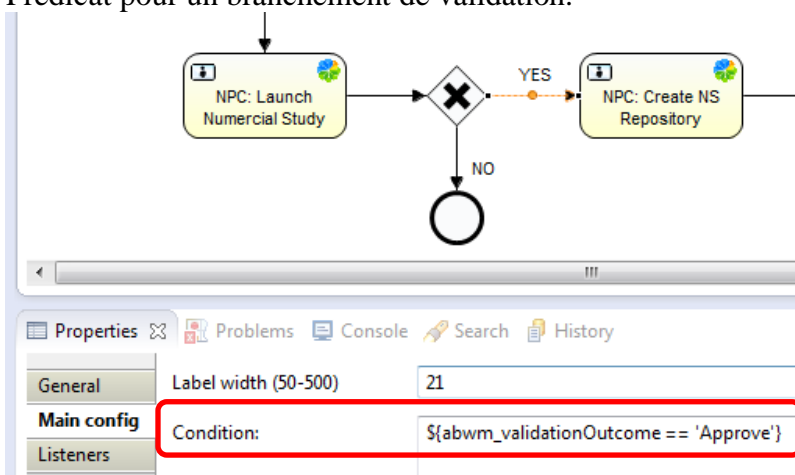


Figure 40 - Exemple 1 de prédicat sur un branchement exclusif.

Prédicat pour un branchement de rejet.

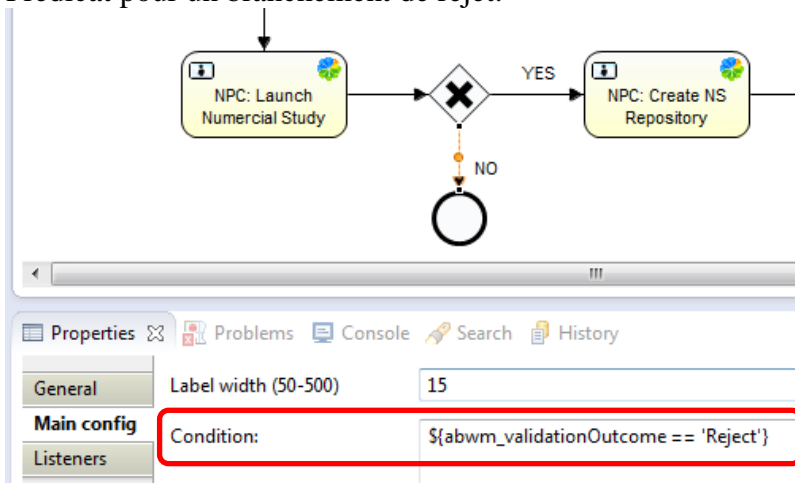


Figure 41- Exemple 2 de prédicat sur un branchement exclusif.

L'ensemble des fonctionnalités proposées par Activiti n'a pas été exploité dans la réalisation du prototype du projet *Acoustic*.

Après l'implémentation de la logique directement dans le XML-BPMN 2.0 du processus, avec l'aide de l'outil Activiti Designer, le processus est prêt pour être exécuté par le moteur de workflow. Pour qu'il soit possible d'exécuter un processus il doit être déployé dans le moteur. La section suivante va arborer comment il est possible de procéder.

L'implémentation d'un processus sur un moteur de workflow est contrainte technologiquement par le moteur et de sa facilité de prise en main. D'une part par sa couverture de la spécification BPMN 2.0 puis de la manière dont il est possible d'implémenter l'assignation des *User Tasks*, l'association de ces éléments avec une IHM, l'accessibilité aux variables d'instances de processus et l'injection de routines logiques pour automatiser les branchements. Le moteur doit fournir les outils nécessaires permettant de faciliter son implémentation et de contrôler sa qualité.

IV.4.2 Exécution dans Alfresco

Le processus NS-P1 est exécutable par Activiti mais il doit être capable de s'intégrer dans Alfresco.

Avant de pouvoir le déployer il reste des configurations à faire pour que, d'une part, les données qui seront utilisées par les formulaires des tâches utilisateurs puissent être traitées et d'autre part, la création de la configuration des formulaires Share référencées par leur attribut **formKey** dans la définition XML-BPMN 2.0 du processus. Les groupes définis dans l'assignation des tâches utilisateurs doivent être présents dans Alfresco ou créés.

Un plan de séquences d'étapes pour parvenir à l'exécution d'un processus dans le moteur Activiti d'Alfresco par adaptation du XML est présenté dans la Figure 42 et sera détaillé dans la suite de cette section.

1. **Process Definition** (ou définition de processus) : La définition du processus est adaptée et les possibles classes Java de délégation ou bibliothèques tierces sont identifiées.⁴³
2. **Task Model Definition** (ou Modèles de tâches utilisateurs) : configuration d'Alfresco permettant l'exécution de processus
3. **Share Web Client Configuration** (ou Configuration des IHM) : configuration des interfaces hommes-machines utilisateurs et leur ergonomie.
4. **Resource Bundle Configuration** (ou Configuration des ressources) : configuration de la localisation du processus.

⁴³ Cette étape a été présentée dans la section précédente IV.4.1

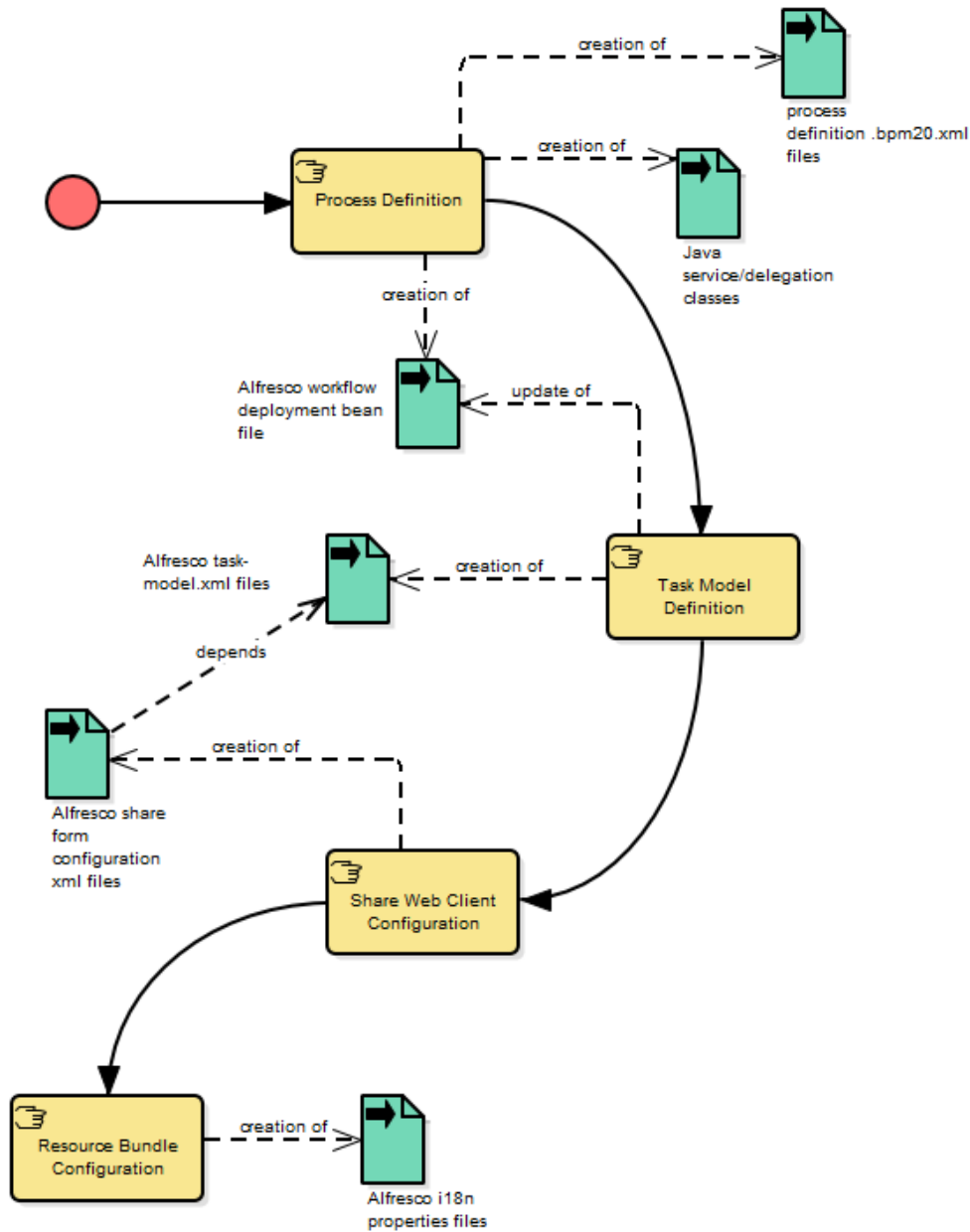


Figure 42 - Procédé d'exécution de processus BPMN 2.0 sur Alfresco.

Il existe plusieurs moyens pour déployer une définition de processus dans Alfresco et pour ce projet j'ai opté sur l'utilisation d'un *workflow deployment bean*. Le principe est de pouvoir déposer un fichier d'extension d'Alfresco dans un répertoire et d'automatiser le déploiement de la définition de processus. Ce principe est automatisable et ne nécessite pas d'intervention humaine et rentre dans le cadre d'une approche d'intégration continue et de déploiement continu⁴⁴ auquel j'adhère et sur laquelle j'étudie actuellement la mise en place sur les projets de commerce électronique chez X2i.

⁴⁴ L'intégration continue et le déploiement continu sont des principes permettant la mise en production automatisée de logiciels ayant passé toute la chaîne de contrôle de qualité automatisé. C'est une étape d'industrialisation des processus DevOps. L'article suivant vulgarise bien ce principe (16).

Alfresco a besoin de 3 artefacts pour pouvoir déployer des définitions de processus, ces artefacts font référence à la définition de processus au format BPMN 2.0.

- Modèles de tâches utilisateur (ou *Task Model*)
- Configuration des IHM (ou *Share Web Client Configuration*)
- Fichiers de localisations (ou *Resource Bundle Configuration*)

IV.4.2.1 Modèles de tâches utilisateur

La définition des modèles de tâches utilisateur est proche du principe de la définition des modèles de données⁴⁵ dans Alfresco (13). Cela correspond avec une modélisation des métadonnées et relations d'un élément de contenu dans Alfresco. Cette fonctionnalité est aussi puissante que complexe. Un modèle de données sous Alfresco est un système qui permet de décrire les données persistées dans l'entrepôt des données⁴⁶. C'est un dictionnaire des types de données, des propriétés (métadonnées), des contraintes, des associations, des aspects de contenu et des comportements personnalisés.

Chaque entrée du dictionnaire est utilisée pour décrire un type de contenu, par exemple la représentation d'un répertoire ou d'une typologie de documentation d'entreprise. Les types de contenus sont équivalents à des classes dans la modélisation objet comme l'indique la Figure 43 :

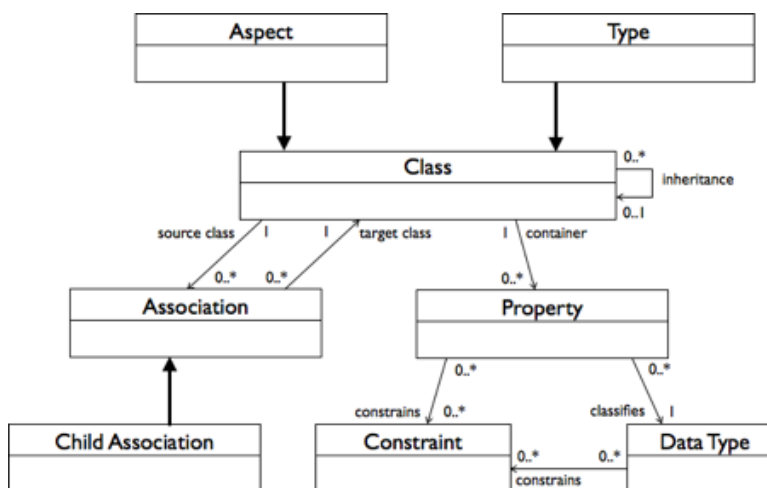


Figure 43 - Méta modèle du modèle de contenu sous Alfresco (14).

Un modèle de *User Task* est un descriptif des données de chaque *User Task* présent dans la définition de processus. Chaque description doit contenir un nom et un titre, des propriétés et des relations (par exemple les documents associés à une tâche). Ce descriptif est utilisé pour générer les interfaces utilisateurs permettant de visualiser et de gérer les différentes tâches d'une instance de processus. Le descriptif se base sur le dictionnaire de modèles de données d'Alfresco étendu pour la définition de modèles BPMN et des processus.

Le diagramme de la Figure 44 montre la projection des modèles de données du processus NS-P1 faite en phase de modélisation sur les modèles de *User Task* Alfresco.

⁴⁵ « Content Model » en anglais.

⁴⁶ « Alfresco Repository » en anglais.

Name: abwm-task model
 Package: workflow-specific content model
 Version: 1.0
 Author: tvercrusse

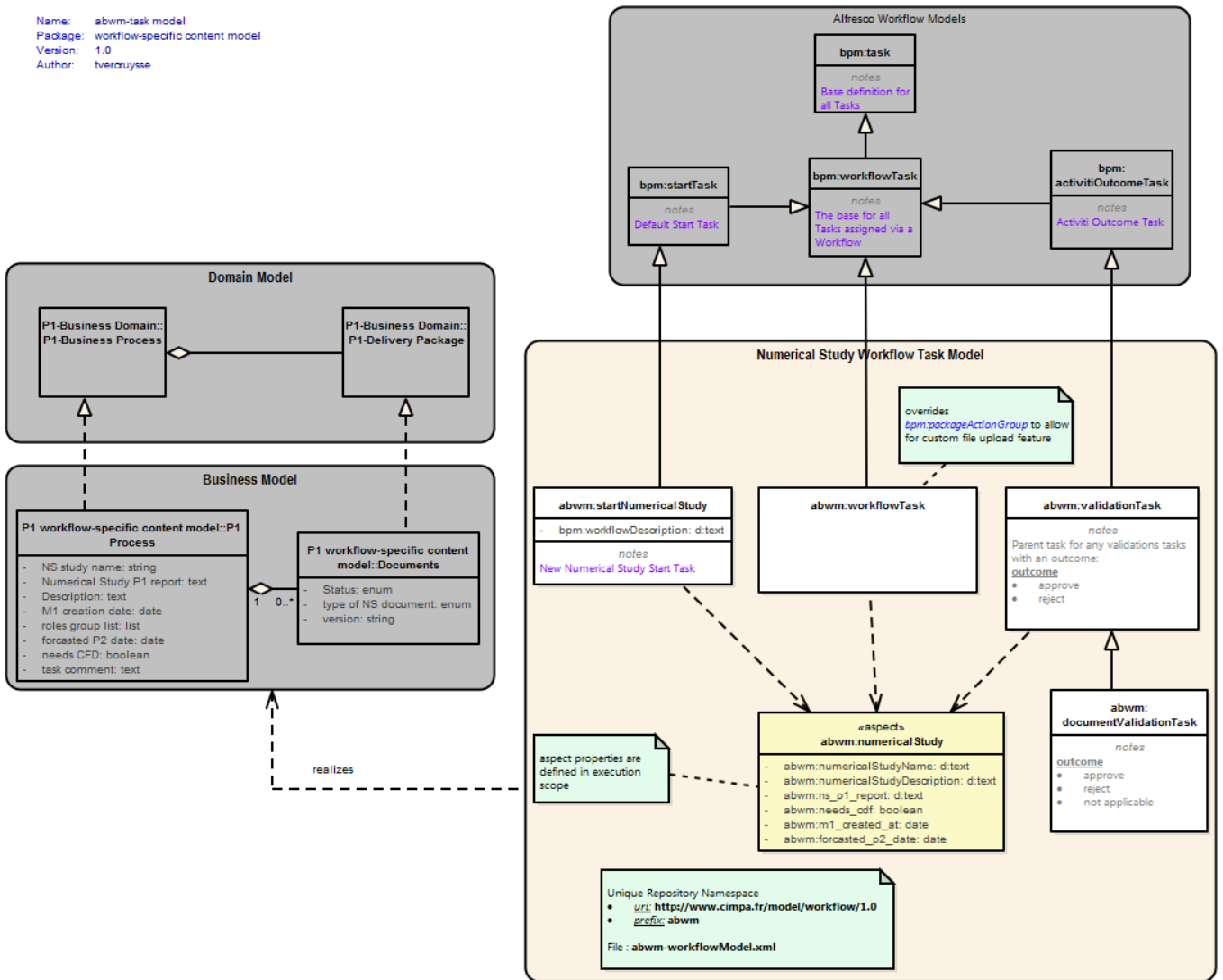


Figure 44 - Définition du Task Model Alfresco du processus NS-P1.

Quand nous souhaitons définir nos propres modèles de tâches utilisateurs dans Alfresco il faut identifier quel modèle existant sera réutilisé et étendu. Par exemple un *Start Task* doit étendre un **bpm:startTask**. Chaque *User Task* avec une issue décisionnelle doit étendre le **bpm:activitiOutcomeTask** ensuite les autres *User Tasks* doivent finalement étendre le **bpm:workflowTask**.

J'ai utilisé le concept des *Aspects Alfresco* afin de pouvoir mutualiser des données entre les différentes tâches pendant l'exécution d'une instance de processus. Les valeurs transmises par les *Aspects* sont globales, elles font partie des variables de processus, pour une instance de processus, tandis que les valeurs transmises par les modèles de contenu doivent être explicitement passées au niveau des variables de processus. Cette contrainte à nécessité une adaptation des scripts et classes Java construits. Le constat est que la partie sous-jacente, exécutable, d'un processus BPMN doit être aussi adapté afin de pouvoir être exécutable sur Alfresco en fonction de la définition du modèle de tâches utilisateur.

La configuration du *bean de déploiement* référence cette définition de modèle de *User Task* spécifique à une définition de processus. Nous avons utilisé un espace de nom unique sur ce projet : **abwm**.

Un deuxième constat sur l'implémentation d'un processus métier sur Alfresco est que la définition de spécialisation du modèle de donnée Alfresco est lourde et chronophage. A chaque adaptation du modèle des itérations dans la définition de processus BPMN est nécessaire au niveau exécutable. Ce projet expose un seul processus mais sur des projets de grande envergure avec plusieurs processus d'entreprise cela m'inquiète. Une modification à un endroit implique des modifications dans plusieurs. Afin de garantir au moins une non-régression je propose de mettre en place des tests unitaires de processus sur des projets de grande envergure ou les projets nécessitant des mises à jour régulières. En réalité quand un processus change le cycle BPM⁴⁷ doit être re-parcouru depuis le début comme indique la discipline BPM. Sur cette solution un responsable métier ne peut pas modifier un processus BPMN sans potentiellement impacter tout le système, idéalement je trouve que ça manque de souplesse et que c'est une des contraintes d'une informatisation de processus non négligeable à prendre en compte dès le début d'un projet.

Afin d'être efficace, robuste et évolutif cette phase de modélisation des modèles de données Alfresco doit être faite en connaissance de cause et mérite une attention particulière.

Un des besoins du projet *Acoustic* est de pouvoir centraliser les documents du processus et d'avoir une visibilité sur l'avancement de leur état. Alfresco est un gestionnaire de contenu électronique et j'ai choisi d'exploiter cette puissance.

Chaque document du processus a une machine d'états associé : *Draft, Waiting Approval et Published*. Une typologie de documents a été faite en phase de modélisation des attachements documentaires et chaque document est classifié en fonction de son type : contexte géométrique, contexte acoustique, contexte aérodynamique et contexte métier.

Suite à des échanges pendant des réunions de travail nous avons opté sur l'implémentation de la projection présente dans le diagramme de la Figure 45, le client souhaitait pouvoir manipuler ces documents dans des répertoires de l'entrepôt documentaire d'Alfresco.

⁴⁷ Cf. section Cycle BPM page 22 de ce document.

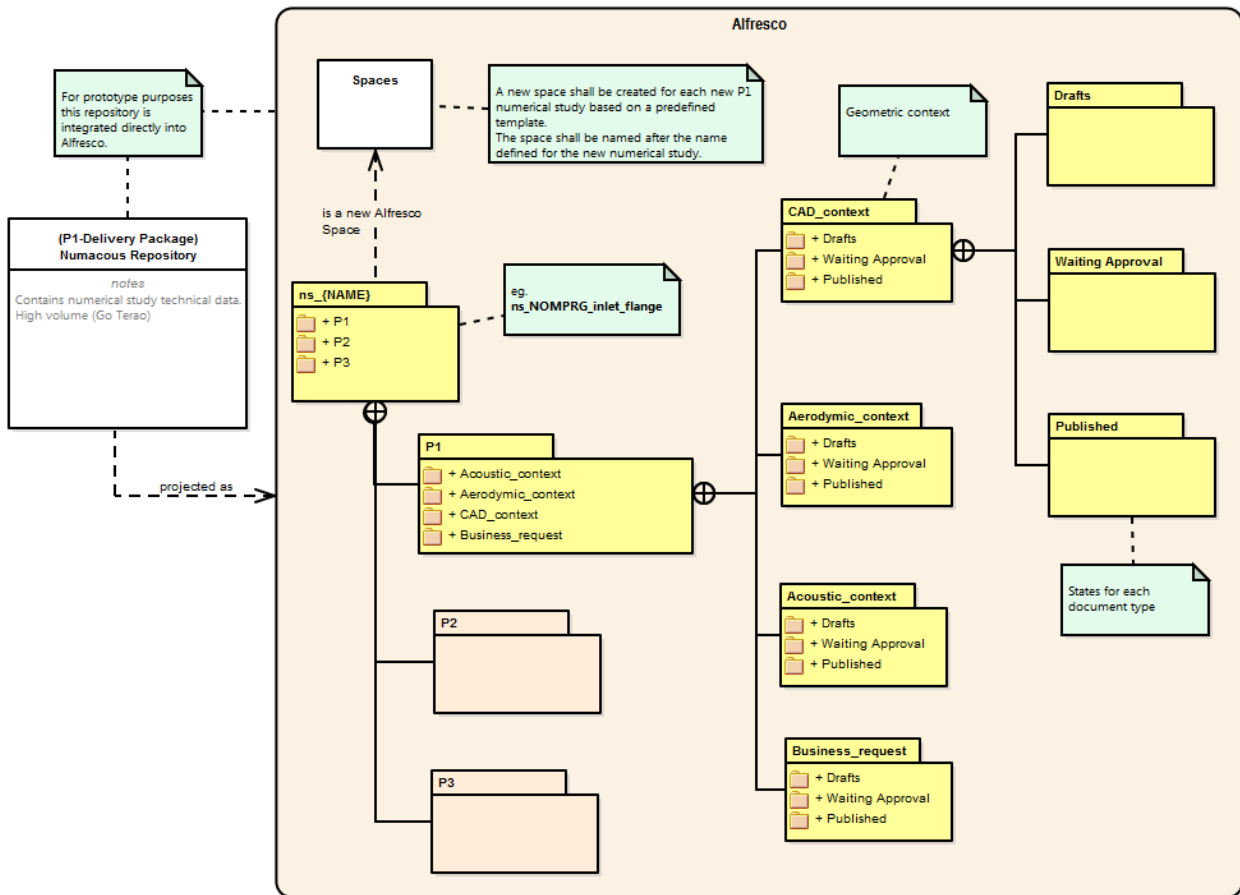


Figure 45 - Projection des attachements documentaires du processus NS-P1 sur l'entrepôt de documents Alfresco.

Ce cas illustre une des puissances de l'implémentation de processus sur des logiciels, en complément du séquençage automatique de tâches et l'utilisation de formulaires assignés, il est possible avec une injection de logique de manipuler directement le logiciel.

L'implémentation de processus dans des logiciels permet une manipulation directe des données et les couches applicatives du celui-ci. L'implémentation d'un processus permet une synergie avec le logiciel qui l'exécute.

IV.4.2.2 Configuration des IHM

Dans la section précédente nous avons associé chaque *User Task* avec une IHM de gestion de tâche. Dans Alfresco, il faut créer une configuration *Share Forms* pour chaque formulaire décrivant les différents champs qui seront affichés et les références vers le modèle de donnée qui sera utilisé pour gérer les données des formulaires. Cette configuration sera déployée automatiquement. (15)

Le fonctionnement entre le moteur Activiti et Alfresco est le suivant : un utilisateur souhaite afficher une tâche sur laquelle il est assigné. Le moteur de workflow Activiti demande à la couche d'affichage Share de récupérer la configuration de formulaire et de chercher la configuration correspondante avec la propriété *formKey* du *User Task*. Ensuite Share affiche le formulaire avec les données provenant du modèle de contenu qui peuvent contenir des variables d'instances du processus en cours d'exécution. L'utilisateur traite le formulaire puis Share envoie les données au moteur Activiti qui poursuit l'exécution du processus.

Un point fort de l'exécution des processus avec Activiti est la possibilité de suivre graphiquement l'avancement de l'exécution d'une instance de processus. C'est un moyen de contrôle puissant qui a été bien accueilli par l'équipe du projet.

Le diagramme de la Figure 42 précédente indique une dépendance forte entre la configuration des formulaires Share et la définition des modèles de données Alfresco du processus. Toute modification du modèle de données lève un risque que la configuration des formulaires Share soit impactée. Le problème, comme pour la définition du modèle de données Alfresco, est que même si une définition de processus est versionable, chaque déploiement de processus porte un numéro de version et chaque instance porte une référence vers le numéro de version de la définition de processus à partir duquel elle a été instanciée, ces configurations ne le sont pas et cela peut avoir des impacts sur les anciennes instances.

Ces phases doivent pouvoir assurer un *minima* de rétrocompatibilité⁴⁸ sur l'évolution d'un processus quitte à créer une version majeure du processus avec un nommage différent si les modifications sont top importantes.

IV.4.2.3 Configuration des ressources

Comme toutes applications utilisant des IHM, Alfresco Share est localisable⁴⁹. Les processus portent une séquence d'activités utilisateur, des logiques de branchements et des données. Ce sont des informations pouvant être présentées à l'utilisateur dans un langage précis.

Afin de rendre les formulaires Share et les composants Alfresco d'affichage de contenu provenant d'une instance de processus localisable, une configuration des ressources est faite. Une configuration de ressource fournit tous les messages lus par un humain affiché dans les différentes IHM de gestion de processus. Ces messages incluent les intitulés des tâches, nom des propriétés, nom des choix des issues de formulaires, etc. Afin de pouvoir implémenter un processus dans Alfresco, on se base sur le mécanisme de fichier de langue sous la forme de paquet de ressources localisables (ou *resource bundle*) standards. Ces fichiers de langue sont au format de fichiers « *properties*⁵⁰ » Java.

Ces fichiers *properties* sont déclarés dans le *bean de déploiement* du processus et déployés en même temps que le processus.

⁴⁸ La rétrocompatibilité, ou compatibilité descendante, désigne la compatibilité d'un logiciel vis-à-vis de ses versions antérieures.

⁴⁹ La localisation est la technique qui permet de créer des logiciels multi-langues. La langue étant sélectionnée le logiciel cherche des correspondances dans des fichiers de langues.

⁵⁰ .properties est une extension de fichier essentiellement utilisée en Java et qui permet aux technologies compatibles de stocker les paramètres de configuration d'un logiciel (17)

IV.4.2.4 Déploiement

Les fichiers de définition de processus XML-BPMN2.0 peuvent être déployés individuellement et aussi sous forme d'une archive compressée avec l'extension **.bar** (pour *Business Archive*⁵¹). La méthode de compression est identique à celui des archives au format zip. Cette archive rend la définition de processus portable et transmissible via des systèmes de réseaux vers des logiciels contenant un moteur d'exécution de workflow.

Dans Activiti une archive **.bar** contient tous les processus du projet sous la forme de définitions XML avec l'extension **.bpmn20.xml**. Ainsi que des images de chaque processus. Mais pas que, cet archive peut contenir d'autres fichiers associés à la définition de processus : configuration de formulaires, règles métiers⁵², classes Java de délégation utilisées par le processus et des fichiers de configuration pour le logiciel cible.

Une archive **.bar** est considérée comme une unité de déploiement et de ce fait peut rentrer dans une discipline de déploiement automatique.

Pour une implémentation et exécution dans Alfresco les fichiers de configurations de modèles de contenus, *bean de déploiement*, configuration de formulaires, composants UI spécifiques, paquets de ressources, etc. doivent être ajoutés à cette archive puis transmis à Alfresco pour intégration.

J'ai opté pour une automatisation de ce déploiement en utilisant l'outil *Apache Ant* et mis en place un script de déploiement sur le projet. L'avantage, en développement, est de pouvoir effectuer des modifications puis de les déployer rapidement dans Alfresco pour les tester sans perdre de temps ou commettre des erreurs. La Figure 46 montre les différents fichiers de déploiement et leur organisation dans Activiti Designer pour le projet *Acoustic*.

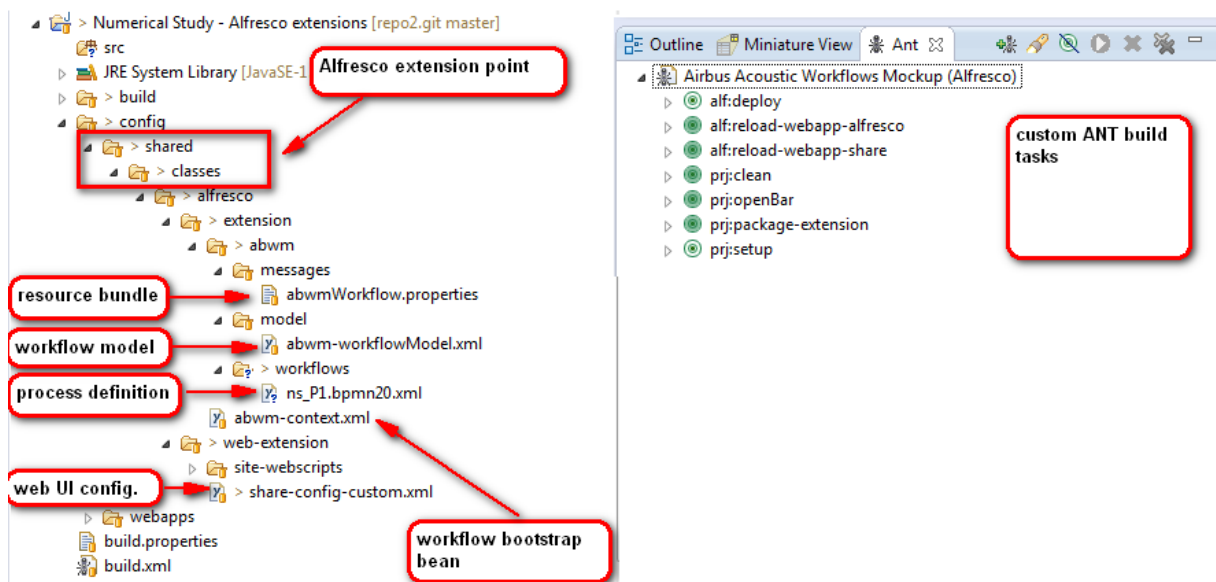


Figure 46 - Configuration de l'espace de travail dans Activiti Designer.

⁵¹ A la manière des archives .jar pour Java Archive.

⁵² La gestion des règles métiers est une fonctionnalité d'Activiti mais n'a pas été sujet d'étude pour le projet *Acoustic*.

IV.5 Bilan

L'implémentation d'un processus est plus complexe que sa modélisation. Le choix d'un moteur de workflow adapté est fondamental et une connaissance technique de la solution cible, essentielle. A chaque tâche utilisateur⁵³ d'un processus, il faut concevoir sa représentation visuelle pour les interactions humaines ainsi que la structure des données qui y seront manipulées. Il est crucial de comprendre comment le moteur de workflow implémente la spécification BPMN 2.0 et comment il la spécialise. Puis ensuite par quels moyens il sera possible d'implémenter et exécuter ce processus dans le logiciel final. La méthode présentée n'est pas applicable pour tous les cas particuliers et toutes les plateformes technologiques mais elle donne un point de vue sur les attentes, contraintes, risques et les parties sur lesquelles il faudrait être vigilant.

Les principes du cycle de qualité BPM indique clairement qu'un processus évoluera afin qu'il puisse être optimisé en accord avec les indicateurs de performances posés par l'entreprise pour laquelle l'informatisation de processus a été réalisé. L'implémentation de processus doit être ouverte pour modification et contrôlable afin d'assurer un niveau de qualité interne acceptable au niveau de la fiabilité et la non régression du processus.

L'implémentation est très technique et doit être abordée avec toute la rigueur de projets informatiques plus « classiques ». La documentation des modèles de données et des IHM doit être réalisée puis communiquée. Il faut s'assurer que cette documentation est lue et comprise. Le fonctionnement d'un processus doit être expliqué aux équipes de réalisation sachant que dans plupart des cas elles ne sont pas instruites à la lecture d'un diagramme de processus formalisé.

L'équipe technique de réalisation doit garder à l'esprit que conceptuellement une définition de processus doit être éditée par une personne non-technique. Des activités peuvent être remplacées par des nouvelles ou les existantes réutilisées, des nouveaux branchements peuvent apparaître et des anciennes modifiées. L'implémentation doit être faite, dans la mesure du possible, sous la forme de composants graphiques réutilisables et autonomes. Ceci permet de réduire les efforts et les risques de maintenance.

Il est important dans l'implémentation de processus de procéder à des rencontres régulières avec l'utilisateur final ou son représentant. Puisque par définition un processus métier est utilisé dans le cadre d'une entreprise et les acteurs sont humains. Le client doit être au centre d'un projet d'informatisation de processus humain et y participer activement. Les méthodes de gestion de projet agiles sont, à mon sens, bien adaptées.

Concernant le moteur de workflow Activiti et son utilisation avec Alfresco et Share, je suis arrivé au constat que, techniquement, le moteur est assez simple d'utilisation pour des processus sur-mesure et j'ai pris plaisir à réaliser ce projet. L'équipe technique d'Activiti est bien dans le sens où ils répondent aux interrogations rapidement et sont pertinents dans leurs réponses. La documentation technique du moteur est bien faite. Par-contre d'un point de vue utilisation métier la plate-forme Activiti n'est pas ergonomique et sa mise en œuvre dans des projets nécessite une collaboration avec des développeurs.

⁵³ Ou *User Task* en anglais.

J'ai passé une bonne partie de la phase d'étude à monter en compétence en me documentant sur ce moteur et sur ses possibilités pour intégrer le processus NS-P1 dans Alfresco. Au fur et à mesure de mon avancement dans la réalisation du projet, j'ai maintenu un journal de développement. Il est ensuite devenu une procédure réutilisable, rédigée en anglais, d'implémentation de processus humains dans Alfresco avec le moteur de workflow avancé Activiti et a par la suite j'ai eu le plaisir d'apprendre que cette procédure a été référencée par Airbus.

L'implémentation du processus NS-P1 dans Alfresco a eu quelques mauvaises surprises techniques, d'un part par manque de maîtrise de la plate-forme Alfresco et d'autre part par un certain nombre de dysfonctionnements au niveau du moteur Activiti et de limitations essentiellement dues au fait que la version du moteur Activiti embarqué dans Alfresco n'est pas la même que celle distribuée directement. Certaines fonctionnalités essentielles n'étaient alors pas encore disponibles et des contournements ont été réalisés puis documentés dans la procédure.

La montée en compétence pour l'implémentation d'un processus sur cet environnement n'est finalement pas très compliquée mais requière à la fois une bonne compréhension technologique de ce que représente l'exécution d'un processus sur le moteur Activiti et de l'architecture de la couche applicative et infrastructurelle de la plate-forme Alfresco.

Quel serait l'effort nécessaire pour implémenter ce même processus sur un logiciel ayant une couche applicative et infrastructurelle différente ?

Afin de présenter cette problématique et pour finaliser cette section, je vous propose d'observer un diagramme d'architecture montrant les trois couches de l'implémentation de processus sous Alfresco avec le standard de modélisation *Archimate 2*.⁵⁴

Ces trois couches sont conceptuellement les suivantes :

- La couche **métier** dans laquelle nous voyons les produits et les services qu'une organisation propose à ces clients ainsi que les acteurs et les rôles dans cette organisation.
- La couche **applicative** décrit les systèmes et leurs fonctions ainsi que les données échangées.
- Et la couche **d'infrastructure (ou technologique)** dans laquelle nous identifions les ressources matérielles, les réseaux et les logiciels système.

Dans cet exemple, la couche applicative et d'infrastructure forment un **ensemble technologique distinct**, celui de la plate-forme Alfresco combiné avec le moteur de workflow Activiti et la couche de présentation Share Forms. Cet ensemble est présenté dans la Figure 47 ci-après.

⁵⁴ Standard de modélisation supporté par l'OMG permettant de représenter un logiciel sous l'angle de vue de 3 niveaux (ou couches) : métier, applicatif et infrastructure.

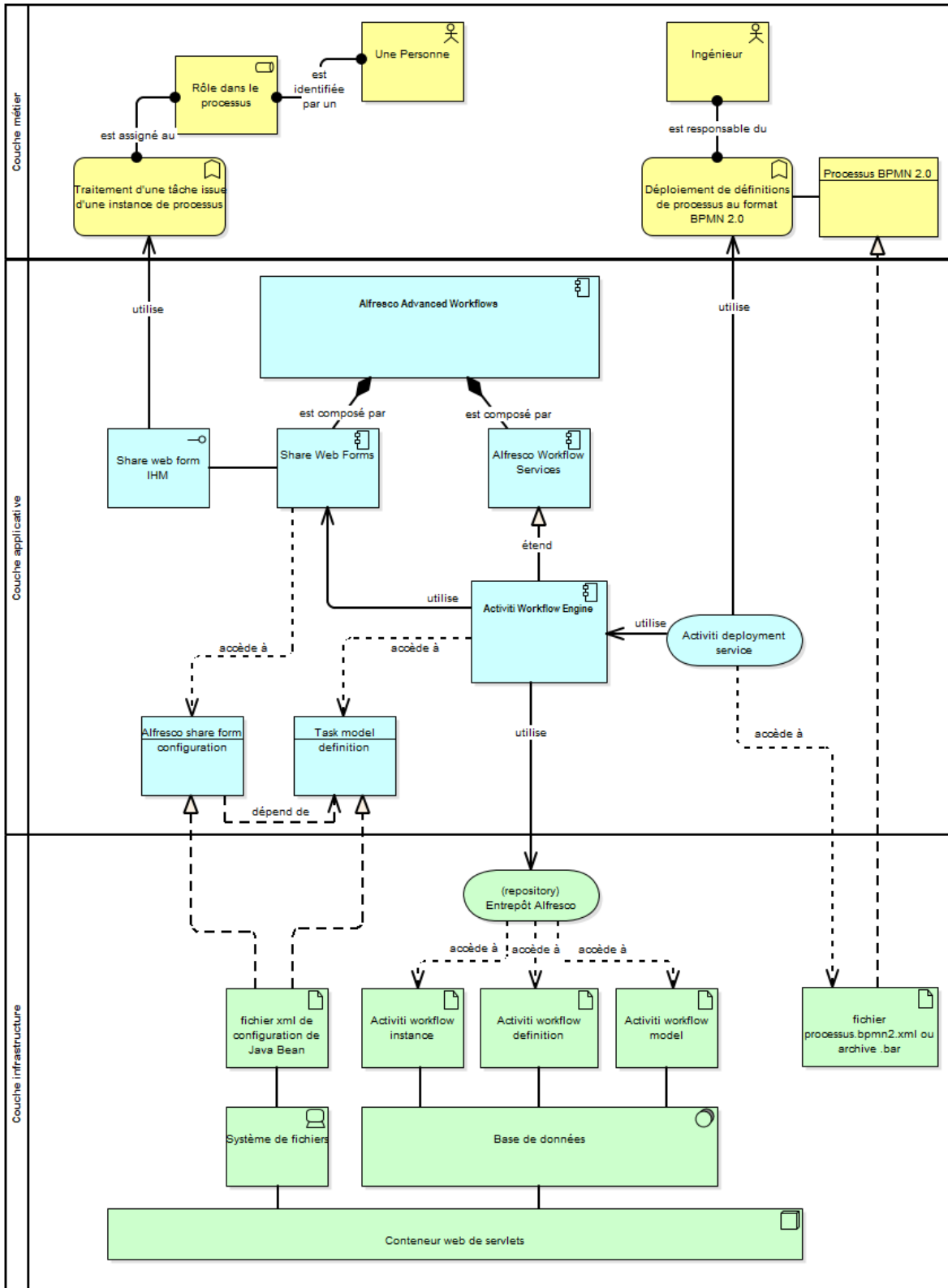


Figure 47 - Architecture en 3 couches de l'implémentation d'un processus avec Activiti et Alfresco.

Imaginons maintenant que les besoins de la couche métier soient maintenus et que l'ensemble technologique applicatif et d'infrastructure soit remplacé par un autre ensemble technologiquement distinct (ou hétérogène).

Nous aurons le besoin d'identifier les composants d'équivalences suivants sur cette solution technologique :

- Moteur d'instanciation et gestion des processus.
- Gestionnaire de formulaires IHM.
- Service de persistance des données liées au processus.
- Service de déploiement de processus.
- Relation entre un rôle et une tâche utilisateur sur laquelle il est assigné.
- Délégation de logique d'exécution.
- Synergie avec le logiciel client.
- Tableaux de bords de suivi d'instances de processus.

Imaginons encore plus loin, une entreprise de grande échelle avec plusieurs parcs informatiques locaux différents contenant chacun des ensembles applicatifs et d'infrastructure distincts sur lesquels la même définition de processus doit être instanciée.

La partie V « Instanciation dans un contexte hétérogène », ci-après, de ce mémoire montre comment un processus métier a été instancié sur un ensemble technologique différent que celui projet *Airbus Acoustic Workflows*.

V Instanciation dans un contexte hétérogène

La méthode de modélisation d'un processus métier s'avère être facilement réutilisable peu importe la nature technologique du projet. La complexité d'aboutir à une vraie interopérabilité d'instanciation est mise en évidence au travers du second projet concernant la modélisation et l'instanciation de processus sur une solution technologique différente.

Je vous ai exposé avec l'exemple d'Activiti que des moteurs de workflow peuvent proposer des API⁵⁵ sur lesquelles il est possible de s'interfacer avec des logiciels clients. Idéalement ce serait une solution d'interopérabilité pertinente comme l'indique la Figure 48. Dès que la définition de processus formalisé en XML est spécialisée pour le moteur de workflow nous avons un composant centralisé exploitable par des systèmes hétérogènes comme présentée dans la Figure 48.

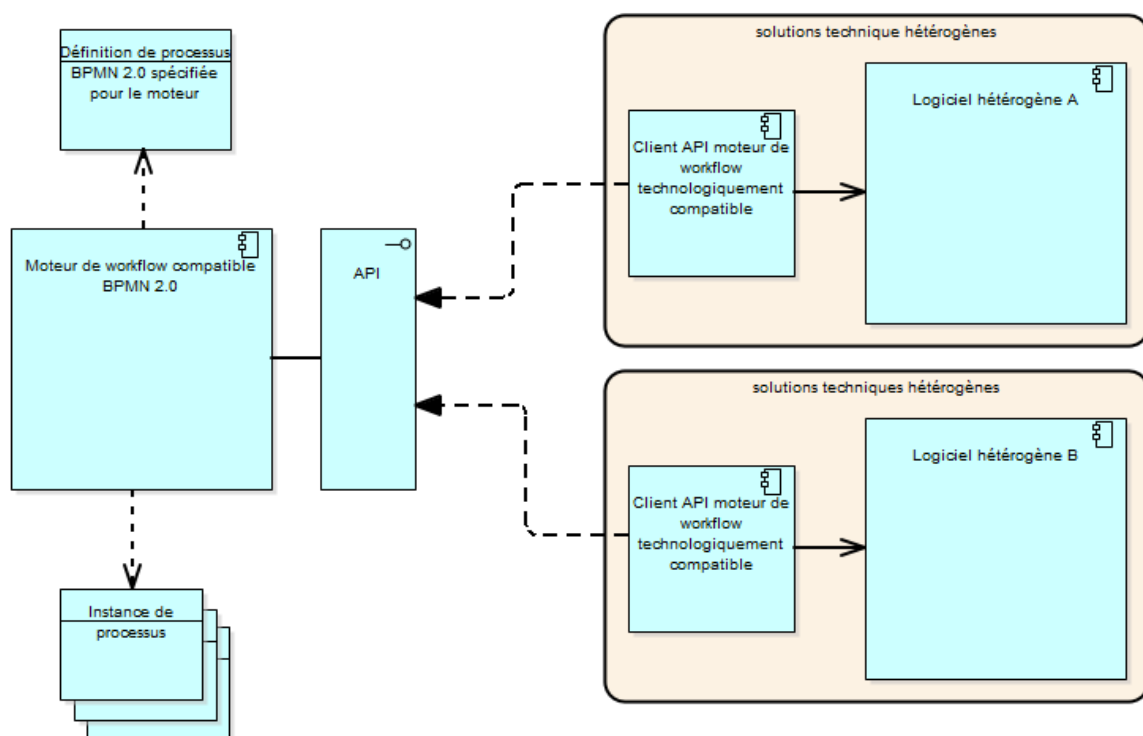


Figure 48 - Exemple d'interopérabilité avec des solutions techniques hétérogènes.

Cependant, cela implique des contraintes et des efforts au niveau de la construction de clients spécifiques sur les solutions hétérogènes. Ainsi que sur la maintenance du composant moteur de workflow qui pourrait ne pas s'intégrer aisément dans un système d'information existant. Le contexte du projet suivant va démontrer pourquoi cette solution n'a pas été réalisable et quelle solution alternative a été préférée.

⁵⁵ L'utilisation d'API a été identifiée sur les deux éditeurs de moteur de workflow BPMN 2.0 les plus populaires à ce jour : Bonitasoft et Activiti.

Par contre je suis convaincu qu'il est possible de réaliser un client suffisamment générique pour une solution logicielle comme Magento et la réutiliser sur des projets différents utilisant la même plate-forme Magento par exemple.

V.1 Cas 2 : Descours & Cabaud et les Circuits de validation



DESCOURS & CABAUD

Suite au projet *Acoustic*, j'ai été mandaté par X2i sur le projet de refonte e-commerce business to business (B2B) sur Magento de l'offre *Descours & Cabaud*. Le groupe *Descours & Cabaud* est leader de la distribution professionnelle des métiers du bâtiment et industriel en France depuis 230 ans avec 590 points de vente répartis entre 4 groupes et 83 filiales comprenant une offre de plus d'un million de références avec plus de 450 000 clients.

Ce projet a nécessité un interfaçage avec 40 ERP pour le calcul des grilles tarifaires des clients B2B et les circuits de validations des demandes d'achats. J'ai été chargé de l'architecture globale et de l'intégration des circuits de validation. Un circuit de validation permet de contrôler la conversion d'une demande d'achat en une commande. Une demande d'achat est faite quand une personne ayant un rôle d'acheteur constitue une liste d'achats pour un chantier par exemple. Cette liste a besoin d'une approbation et seul un décisionnaire peut valider cet achat souvent conséquent. Ce circuit de validation peut passer par plusieurs validateurs avant que la demande d'achat soit convertie en commande. Chaque décisionnaire a souvent l'obligation d'escalader la demande à un autre responsable selon un seuil d'achat.

Techniquement ce projet a été basé sur le framework e-commerce populaire, Magento⁵⁶, utilisant la technologie PHP couplée avec des services tiers. Contrairement au projet *Airbus Acoustic Workflows* cité précédemment, ce projet a une contrainte forte en termes de robustesse du workflow. Dès lors qu'une demande d'achat est convertie en commande elle devient engageante puisqu'elle sera automatiquement traitée par l'ERP et l'encours client sera décompté.

Fort de l'expérience du projet précédent, j'ai choisi d'utiliser la même méthodologie concernant la modélisation de ce processus humain. Le projet a été mis en œuvre en plusieurs phases : phase d'étude, phase de réalisation, phase de livraison et phase d'exploitation. En phase d'étude nous avons organisé des réunions de travail afin de pouvoir itérer sur le besoin des circuits de validation selon le cycle BPM d'amélioration continue d'un processus. La phase de réalisation et d'exploitation a fait ressortir des interrogations et des améliorations du processus selon les principes du cycle BPM.

⁵⁶ Magento est un logiciel de commerce électronique très technique et populaire représentant 2,5 millions de téléchargements. Il est basé sur le framework PHP Zend Framework et vise les projets de grande envergure malgré ses nombreux incohérences et dysfonctionnements, notamment au niveau de ses performances et de son évolutivité.

L'équipe du projet chez X2i a été constitué de la manière suivante :

Chef de projet : Patrice MANSET

Référent Métier: Christophe LeBOT

Référent Technique : Thierry VERCRUYSSSE

Equipe de développement: Abel BRIEN, Jonathan BUREL, Simon LAUBET, Alexandre KHAYRULLIN, Thierry VERCRUYSSSE

Ce projet a une particularité importante, une collaboration avec une équipe de développement externe avec des compétences fortes sur l'intégration de la solution Magento mais inexistante sur des technologies non Magento. Cette équipe externe a pour objectif la reprise de la maintenance du site avec un support ponctuel de X2i.

V.2 Application de la méthodologie

L'application de la méthodologie de modélisation de processus a eu une valeur ajoutée et a pu être appliquée telle qu'elle avait été définie pour le projet *Acoustic*.

Le besoin a été modélisé en domaine et en cas d'utilisations ce qui a permis une fois de plus de revoir les rôles définis ainsi que leurs responsabilités. Comme pour le projet *Acoustic Workflows*, le client a eu la même réaction et constata l'incohérence dans la définition de rôles actuels. Suite à un affinage pendant la réunion de travail nous avons proposé une première version des circuits de validation représentée dans le diagramme de la Figure 49 ci-après.

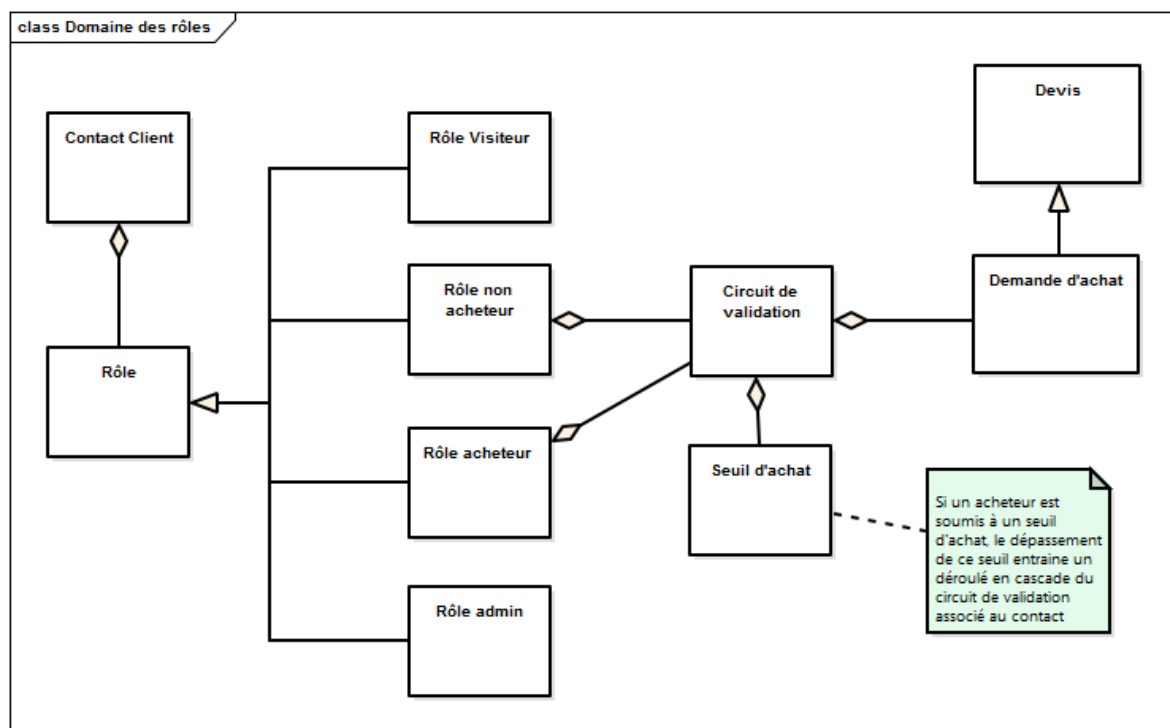


Figure 49 - Modélisation du domaine des circuits de validations.

La définition de ce domaine, par exemple, a permis de désambiguïser le terme circuit de validation qui a été cité à plusieurs reprises aussi avec le nom : circuit d'achat.

La difficulté rencontrée pendant la phase d'étude a été de savoir à quel rôle sera affecté le circuit de validation pour un achat et quel rôle pourra en faire la validation. Le diagramme de cas d'utilisations, réalisé avec la méthodologie ICONIX, de la Figure 50 expose le résultat de cette étude avec le diagramme de cas d'utilisations de la gestion des comptes et rôles.

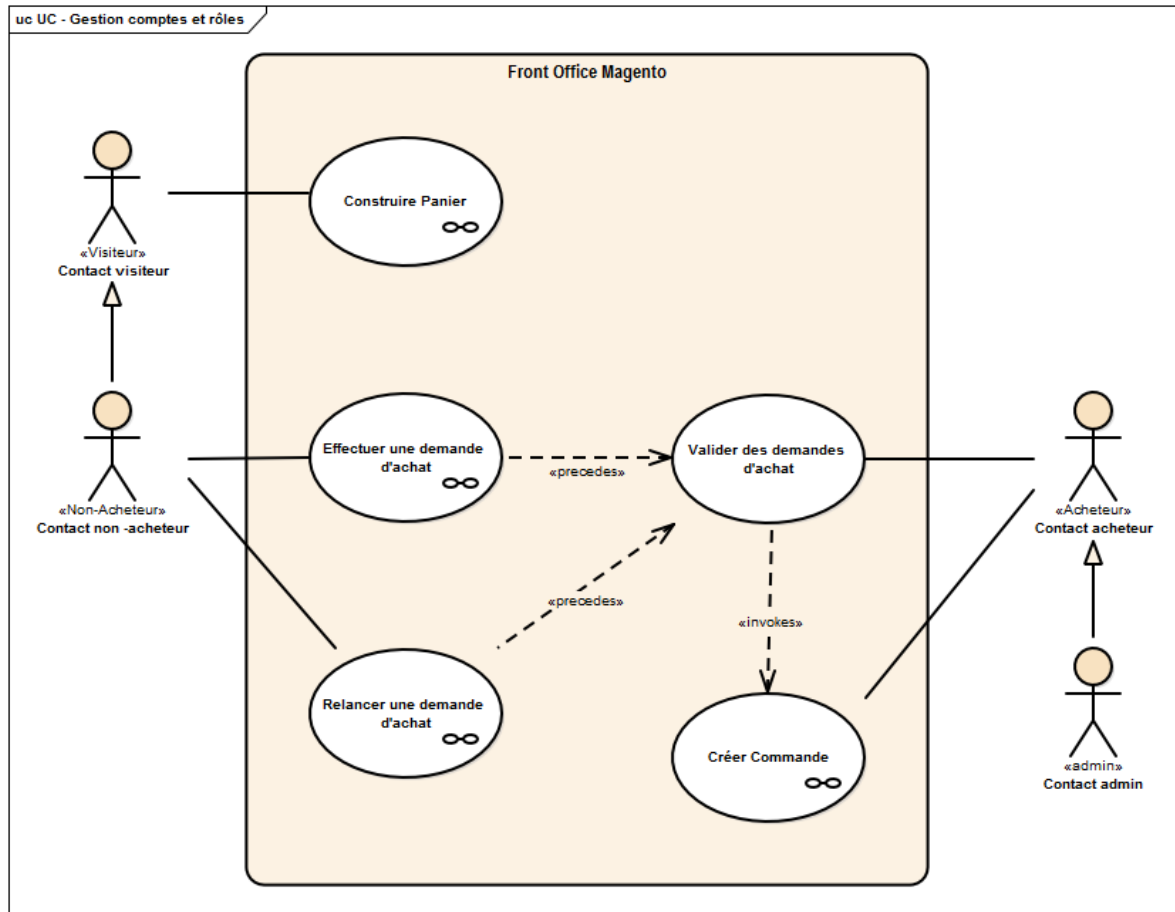


Figure 50 - Diagramme de cas d'utilisations des circuits de validations.

Le client nous avait fourni plusieurs types de circuits afin de couvrir tous les cas d'utilisations imaginées. Après analyse des scénarios, nous avons extrait les observations suivantes.

- Existence d'un pattern commun récurrent.
- Présence de cas extrêmes singuliers et complexes.

Tous les circuits proposés sous forme de cartographie suivaient un pattern commun que je vais vous décrire : le processus de circuit de validation est un processus cyclique qui ne se termine que si un des approbateurs a validé ou refusé le document. Le nombre d'itération du cycle dépend du nombre, variable, de validateurs du circuit. A chaque transition du circuit une notification par courrier électronique est émise. La définition des validateurs d'un circuit de validation et du seuil d'achat le déclenchant est associable à chaque entreprise cliente de la plate-forme.

Chaque cas extrême identifié a été soumis à une contre-analyse de l'utilisation *in situ* de celle-ci et à notre surprise aucune entreprise utilisant l'ancienne plate-forme ne l'avait

utilisé. Nous avons opté sur l'implémentation seule du pattern commun récurrent pour le projet.

La cartographie initiale est présentée dans la Figure 51 suivie par sa formalisation en BPMN 2.0.

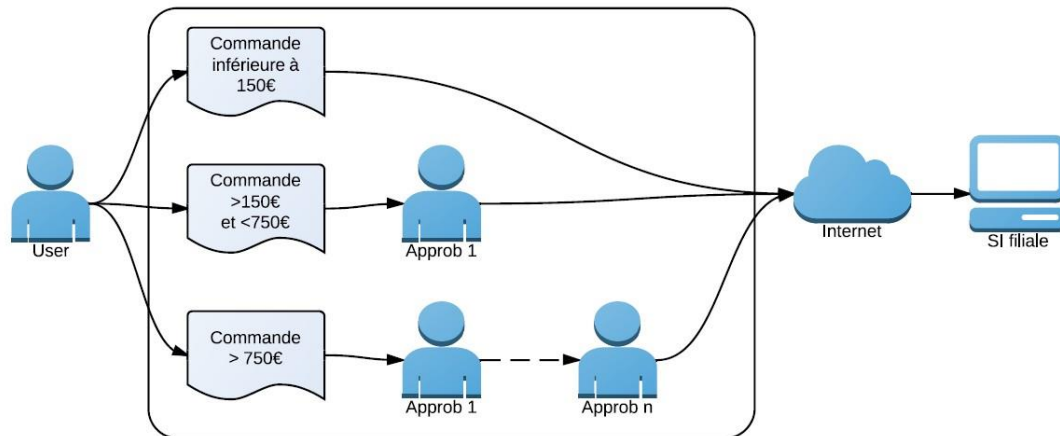


Figure 51 - Cartographie initiale des circuits de validations.

Ce processus a été ultérieurement modélisé suivant le standard BPPMN 2.0 dans le diagramme de la Figure 52.

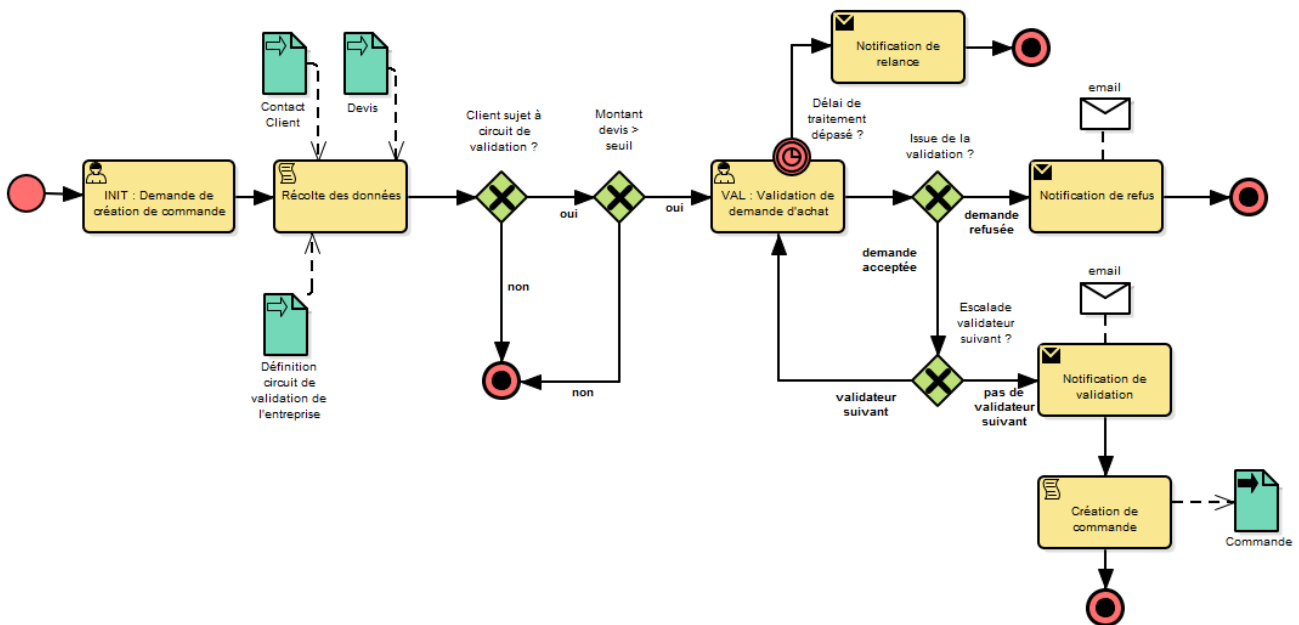


Figure 52 - Modélisation au format BPMN 2.0 du processus de circuit de validation⁵⁷.

La représentation BPMN 2.0 est plus précise, elle indique les entrants, sortants, les branchements possibles, les rôles⁵⁸ et globalement tous les détails du processus. Par contre la cartographie présente une idée, un concept, synthétique.

⁵⁷ Les éléments graphiques, *Lanes*, de la syntaxe BPMN 2.0 représentant les rôles ont été omis pour réduire la taille du diagramme.

⁵⁸ Les rôles de ce processus sont représentés avec les préfixes INIT (ou initiateur) et VAL (Valdateur) des intitulés des deux tâches utilisateur présentes dans la Figure 52.

V.3 Intégration et exécution

La phase d'intégration de ce projet a été complètement différente. D'un part technologiquement Magento ne permet pas d'intégrer nativement un moteur d'exécution de workflow comme le fait Alfresco et il n'existe aucun moteur de workflow compatible BPMN 2.0 codé en PHP⁵⁹. Les technologies des deux plate-formes sont hétérogènes. D'autre part le contexte des projets est différent et la part budgétaire pour l'intégration de la fonctionnalité des circuits de validations trop faible comparé au coût d'intégration d'un moteur Java comme Activiti. D'autant plus que les compétences techniques des équipes n'étant pas orientée Java, l'introduction d'une nouvelle technologie posait un risque majeur sur la maintenance de la plate-forme. J'avais étudié la possibilité d'utiliser Activiti couplé par ses API mais ça a été refusé puisque l'ajout d'un nouveau composant d'infrastructure pour ce couplage ajoutait un coût supplémentaire non budgétisé sur le projet.

Afin d'être adaptable et évolutif, la solution d'implémentation du workflow doit avoir les objectifs suivants.

- Intégrable naturellement dans le framework Magento afin de limiter les risques de maintenance.
- Respecter une terminologie de moteur de workflow afin de faciliter la maintenance et évolutivité du circuit de validation.

V.3.1 Rétro-ingénierie et moteur de workflow sur mesures

J'ai pris le temps de faire une rétro-ingénierie⁶⁰ des composants logiciel d'Activiti pour en extraire une relation terminologique organisée en fonctionnalités majeures. Cela m'a permis de mettre en avant les fonctionnements internes d'un moteur de workflow.

Ces fonctionnements ont ensuite été projetés sur la solution Magento avec le besoin des circuits de validations pour initier l'architecture de la conception détaillée. Le processus informatisé résultant doit réaliser les fonctionnalités suivantes :

- Une définition de processus exécutable.
- Un moteur capable d'instancier un processus et suivre son exécution en persistant son historique d'exécution.
- Chaque instance devrait avoir des variables de processus associées à celui-ci. Dans le cas des circuits de validation, quelques variables importantes sont : la référence vers le document « demande d'achat », le prochain validateur de la chaine de validation, l'état de validation, date de la dernière assignation à un validateur, etc.
- La possibilité d'associer des IHM avec les tâches utilisateurs.
- Une terminologie claire afin d'identifier rapidement les composants logiciels et leur fonctions en corrélation avec le domaine des processus de la Figure 53.

⁵⁹ D'après mes recherches à l'époque du projet. Entre temps une solution <http://www.processmaker.com/> est apparue et m'a l'air prometteur.

⁶⁰ Traduction de l'anglais « reverse engineering »

Cette rétro-ingénierie a aussi mis en avant trois groupes d'éléments nécessaires pour l'exécution de définition de processus :

- Un moteur d'exécution BPMN 2.0.
- Un modèle logiciel d'une définition de processus BPMN 2.0.
- Un analyseur XML-BPMN 2.0.

L'analyseur XML est capable de transformer la représentation XML d'un processus BPMN 2.0 en modèles logiciels sous la forme de définition de processus versionable qui seront manipulés par le moteur. Le moteur est capable d'utiliser la définition de processus pour en faire des instances. Chaque instance a un historique d'exécution et des variables de processus qui peuvent être modifiées pendant son exécution. Il est possible d'ajouter des *listeners* (ou observateurs⁶¹) sur des événements de pré- ou post-exécution permettant d'injecter des scripts de programmation comportementale.

La Figure 53 illustre une représentation du résultat du travail de rétro-ingénierie sous la forme de diagramme de Domaine.

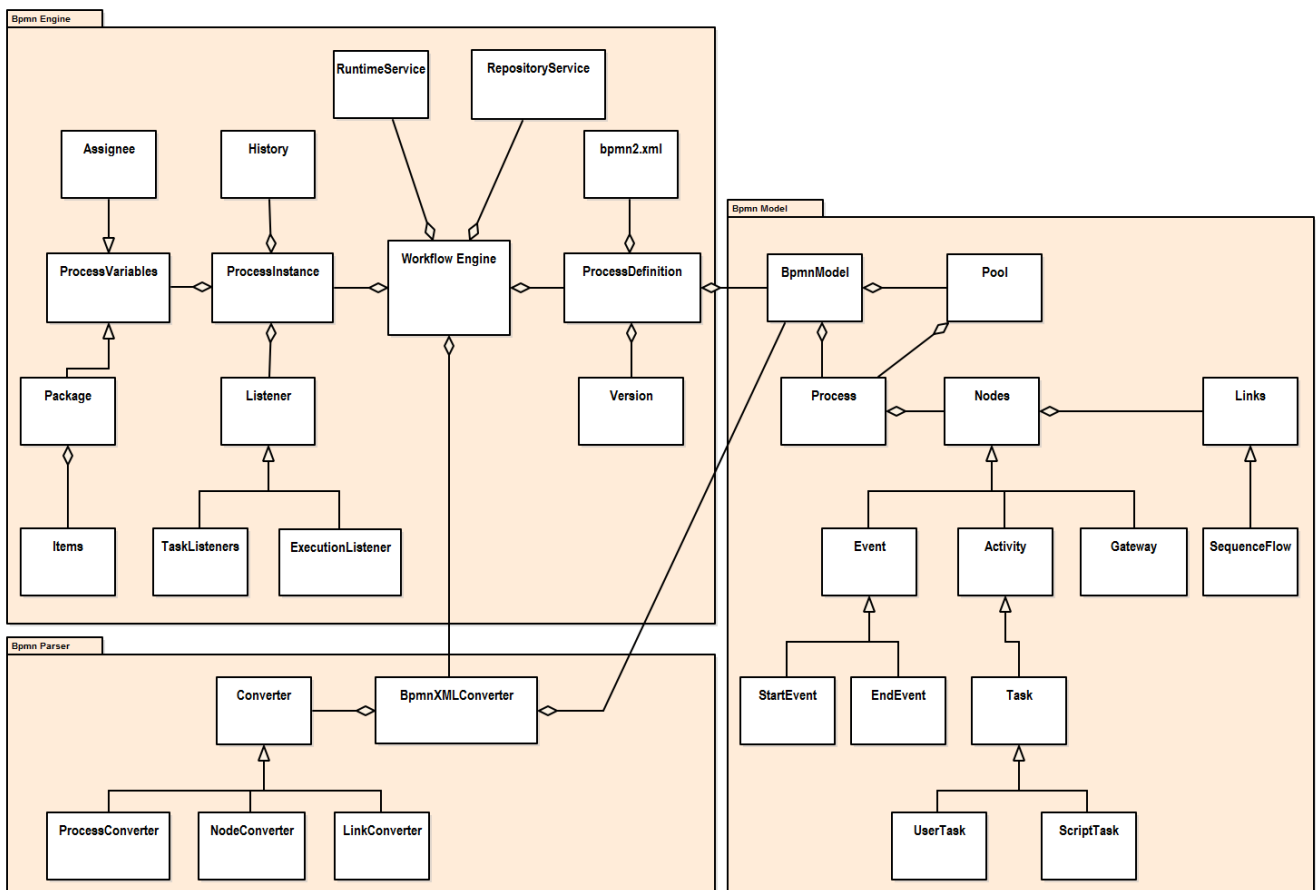


Figure 53 - Diagramme de domaine d'un moteur d'exécution de processus BPMN 2.0.⁶²

⁶¹ Selon le patron de conception logicielle GOF comportementale « Observateur »

⁶² Ce diagramme est non exhaustif pour faciliter sa lecture.

Avec ce schéma il a été possible de se concentrer sur le moteur puisque la formalisation BPMN 2.0 du processus a été validée sur ce projet. Contrairement à l'objectif du prototype réalisé pour Airbus, le moteur ne devait pas être générique parce que ce projet n'avait l'utilité que d'un seul processus. Ayant opté d'être pragmatique, j'ai réalisé une architecture sur mesure sous la forme d'un composant non adhérent avec le reste de la plate-forme. Une contrainte d'architecture a été posée pour qu'aucun autre module de la plate-forme n'ait une dépendance vers ce module. Ce composant a l'avantage d'être réutilisable sur d'autres projets Magento⁶³ avec des besoins similaires en B2B.

La définition de processus permet de persister une configuration d'un circuit de validation et de l'associer à une entreprise de la plate-forme. Chaque définition est « *instanciable* » par le moteur d'exécution et des variables d'exécution lui sont associées. Si la définition est modifiée, l'instance maintient sa définition d'origine. L'initiateur est affecté ainsi que la demande d'achat (le document) et le premier validateur de la chaîne de validation est notifié. L'événement déclencheur d'un circuit de validation est le rôle de l'initiateur quand il tente d'effectuer un achat sur la plateforme ou si le montant d'achat dépasse un seuil défini pour l'entreprise. Plusieurs écrans ont été réalisés permettant de configurer une définition de circuit, de validation et d'en faire la validation, de suivre les demandes en cours d'exécution et les demandes de validations à traiter. Un système automatique de relance a été mis en place pour notifier un validateur qu'une tâche de validation est en attente. Afin de rendre le processus robuste un administrateur de l'entreprise peut prendre la main sur l'avancement de validation et en forcer l'issue.

La Figure 54 montre un extrait de des classes ULM de ce composant (ou module Magento).

La classe *Observer* (à gauche) est responsable de faire la demande à la class *Engine* d'instancier une nouvelle *Instance* à partir d'un objet de la classe *Définition* en fonction du contexte. La Classe *Engine* lui passe un objet *Sales_Order* représentant la demande d'achat.

J'ai fait le choix de projeter les variables d'exécution et la définition de processus dans deux classes. Les différentes données qui représentent une définition de processus de demande d'achat est maintenue dans la classe *Définition*. Et les données d'exécution d'une instance dans la classe *Instance*. Le moteur (ou la classe *Engine*) est capable de *sérialiser* la définition et ainsi de persister la version utilisée pour une instance donnée. Toute la logique d'exécution est portée par la classe *Engine* en fonction des données présentées par la classe *Instance*.

Cette implémentation d'un moteur de workflow est « sur mesure » et ne permet pas l'exécution d'un XML-BPMN 2.0 à proprement parler.

Le système est très élémentaire et évolutif puisque l'implémentation classe *Engine* est remplaçable par un système plus puissant qui pourrait déléguer la logique d'exécution à un moteur de workflow Activiti par exemple en passant par des API REST. A la définition XML-BPMN 2.0 nous pourrions ajouter des observateurs Activiti qui iront récupérer les

⁶³ La plate-forme Magento est dotée d'une architecture modulaire extensible.

données persistées à travers les classes *Definition*, *Sales_Order* et *Instance* par le biais de ces API. La définition XML du processus initial devra être adaptée en fonction sans perdre son sens originel.

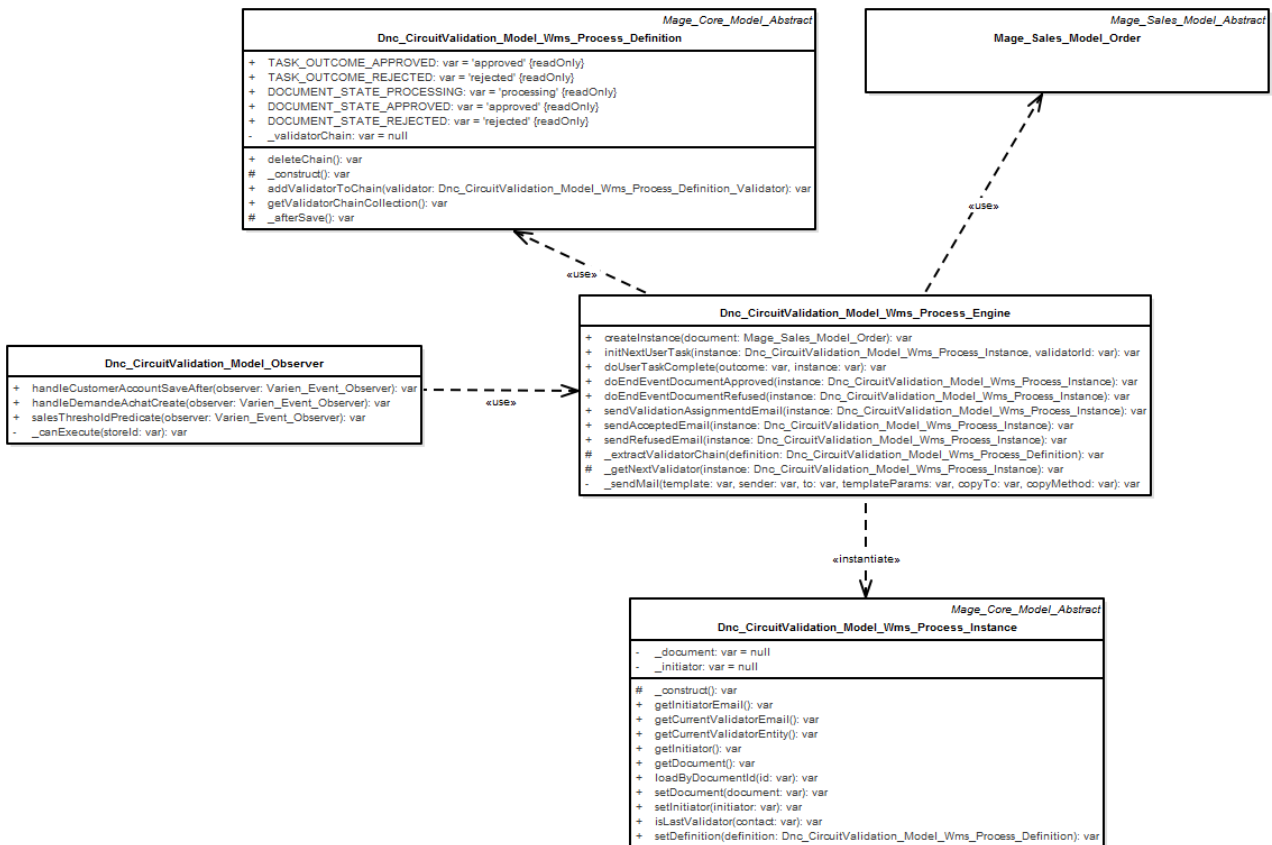


Figure 54 - Diagramme de classes UML du moteur de workflow circuit de validation

V.3.2 Développement piloté par les tests

Etant donné que la fonctionnalité des circuits de validations est critique, le développement a été piloté par des **tests unitaires** afin de garantir un bon niveau de qualité et de non-régression. Les tests unitaires ont été alimentés par un jeu de données (ou *fixtures*) couvrant les cas nominaux et les cas extrêmes d'exécution. Un système de contrôle automatisé avec le logiciel Jenkins a été intégré au projet. Mon objectif est que le module soit portable sur d'autres projets Magento et les test-unitaires ont été packagés avec le code source du module.

Le fait d'avoir modélisé le processus avec le formalisme BPMN 2.0 a grandement facilité la couverture du processus par des tests unitaires. La représentation graphique d'un processus BPMN 2.0 est assimilable à un graphe d'éléments, proche du graphe de flot de contrôle⁶⁴ utilisé pour tester un programme, il est de ce fait possible de s'assurer que chaque branche du graphe de contrôle sera parcourue au moins une fois pendant

⁶⁴ Un flot de contrôle est une représentation sous forme de graphe de tous les chemins qui peuvent être suivis par un programme durant son exécution. Cette représentation permet d'améliorer la qualité et la couverture des tests unitaires.

l'exécution des tests unitaires. Cela facilite grandement le développement piloté par les tests (ou TDD) et limite les oublis ou les cas non identifiés pendant la réalisation ce qui améliore la fiabilité de l'application.

V.3.3 Interfaces homme machine

La Figure 55 représente l'IHM de configuration d'une définition de circuit de validation avec des seuils d'achat. Une définition à plusieurs propriétés : un nom, une description, un seuil, et une liste d'approbateurs. Chaque instance de processus sera alimentée par une configuration de définition.

Figure 55 - IHM de définition d'un circuit de validation

La Figure 56 représente une tâche utilisateur de validation de demande d'achat avec les options d'issue possibles : « Annuler » et « Valider ». Ici c'est le moteur qui récupère les informations présentes dans un objet de la classe *Instance* cité dans la Figure 54 et qui les fournis au système de vue de Magento. Un clic sur un bouton d'issue de formulaire déclenche l'envoi d'un événement et le moteur continu l'exécution du processus.

L'avantage majeur d'une implémentation d'un processus sur un système plus « ouvert » qu'Alfresco est qu'elle permet une plus forte customisation des IHM de processus.

Mon compte

- Historique des commandes
- Demandes d'achat
- Demandes en cours
- Devis
- Panier favoris
- Utilisateurs
- Adresses de livraison
- Circuits de validation

Nom	Statut	PU HT	Quantité	Total HT
3M bricolage et batiment - Réf: 00 82 09 Seuil de différence de niveau de 2,70m inox Codag : 30523954	200 expédiée(s) 200 en attente(s) 200 annulées(s)	264,52 € HT pr 200 Pièce(s)	3 x 200 Pièce(s)	264,52 € HT
3M bricolage et batiment - Réf: 00 82 09 Seuil de différence de niveau de 2,70m inox Codag : 30523954	Livré(s) et Facturé(s)	264,52 € HT pr 1 Pièce(s)	1 x 1 Pièce(s)	264,52 € HT

Frais de port : Selon les conditions de votre contrat.

Montant total HT : **560,64 €**

TVA : **24,33 €**

TTC : **585,25 €**

Retour à la liste des demandes ↩
Annuler ✖
Valider ✔

Figure 56 - IHM de formulaire de validation d'une demande d'achat avec issue⁶⁵.

V.3.4 Suivi du processus

Selon le cycle BPM présenté précédemment un processus doit être suivi avec des indicateurs de performances pour en juger la valeur ajoutée.

Depuis la mise en production du site de commerce électronique en Octobre 2014 un premier rapport réalisé au mois de Mai indique les chiffres suivants sur des comptes réels (hors comptes de test) : ⁶⁶

- Nombre total de commandes : 2413
- Nombre de demandes d'achat : 230
- Nombre de commandes issues des demandes d'achat : 205

Plus de 9% des commandes sont issues d'une demande d'achat. Le client a dégagé un retour sur investissement correct sur cette implémentation et les retours d'utilisation des clients sont positifs suite aux optimisations faites.

V.3.5 Optimisations

Après l'ouverture aux « betas » testeurs de la plateforme, quelques dysfonctionnements de l'implémentation ont été remontés au niveau des IHM. Les tests unitaires ont couvert le moteur et son exécution mais ne permettent pas de couvrir la manipulation des formulaires. Les correctifs ont été mineurs.

Une évolution a été faite pour ajouter plusieurs seuils d'achats sur une définition de circuit de validation afin d'optimiser l'assignation des validateurs d'une chaîne de validation.

⁶⁵ Les montants indiqués dans la Figure 56 sont *factices* et proviennent d'un jeu de données significatif permettant de tester la plate-forme sans avoir de données réelles.

⁶⁶ En *B2B* le nombre de commandes est plus faible que sur un site de commerce électronique « grand public », mais les montants commandés sont significativement plus élevés. D'où l'intérêt de contrôler ces commandes par des circuits de validations.

Une définition de circuit de validation a un ou plusieurs seuils d'achats auxquels est associé une série différente de validateurs.

V.4 Bilan

Bien que ce projet ait été réalisé avec un moteur de workflow « sur-mesure » qui au final ne permet pas d'exécuter directement une définition de processus BPMN 2.0, il a permis de :

- valider la méthode de modélisation de processus ;
- mettre en évidence les composants requis par un moteur de workflow ;
- démontrer qu'il est possible de modéliser un processus, exploiter sa valeur ajoutée et de l'exécuter sans pour autant utiliser le XML-BPMN2.0 et de dépendre d'un moteur d'exécution complexe à introduire dans l'environnement de la plate-forme cible.
- exposer la problématique d'exécution de processus sur un environnement techniquement différent ;
- révéler l'importance de la réalisation de tests unitaires sur l'exécution de processus.

En théorie il est possible avec le moteur d'exécution de processus Activiti, ou d'autres, de pouvoir instancier un processus pour exécution sur des environnements hétérogènes par l'utilisation d'une architecture permettant une interopérabilité de services comme l'indique la Figure 48. Les différents clients utilisent ces services avec la même interface et implémentent chacun leur spécificités.

La limitation de cette configuration se situe au niveau de l'exécution de tâches automatiques et la manipulation des variables de processus directement sur le client. Si les technologies utilisées par ces clients le permettent, ce serait possible avec des technologies d'invocation de programmes à distances comme le RMI Java⁶⁷ ou CORBA⁶⁸ ou d'autres mécanismes d'interfaces de programmation. Dans tous les cas un effort supplémentaire est requis à la fois côté moteur de workflow pour la délégation et côté client pour l'implémentation de cette interface d'invocation de délégation à distances.

Une autre solution pourrait être de simplement déléguer toute la gestion de l'exécution de processus et des IHM associées au moteur de workflow directement. Ce qui multiplie les applications et interfaces utilisateur, selon les entreprises cette solution peut être difficilement adoptée par les utilisateurs et nécessite des formations.

L'instanciation de processus sur des environnements hétérogènes n'a pas de solutions génériques sur étagères. Mais plusieurs suggestions ou orientations de solutions qui dépendent du contexte de son implémentation. Par exemple si l'entreprise à une gestionnaire de contenu Alfresco, avec Activiti l'instanciation des processus pour exécution est faisable, comme présenté avec le projet *Acoustic*. Si l'entreprise n'a pas la possibilité d'intégrer un moteur de workflow directement un effort est à prévoir dans la maintenance puisque son exécution sera faite sur mesure comme présenté avec le projet *Descours & Cabaud*. Si cette même entreprise a besoin de définir plusieurs processus

⁶⁷ *Remote Method Invocation* : une interface de programmation (ou API) pour le langage Java qui permet d'appeler des méthodes distantes.

⁶⁸ CORBA acronyme de *Common Object Request Broker Architecture*.

d'entreprise qui vont évoluer, une architecture distribuée est à prévoir se basant sur des API et le coût de réalisation sera considérablement plus élevé.

Pour conclure cette partie du mémoire je souhaite vous proposer avec la Figure 57 une architecture distribuée d'exécution de processus d'entreprise sur des environnements techniquement hétérogènes avec le standard de notation *Archimate 2*.

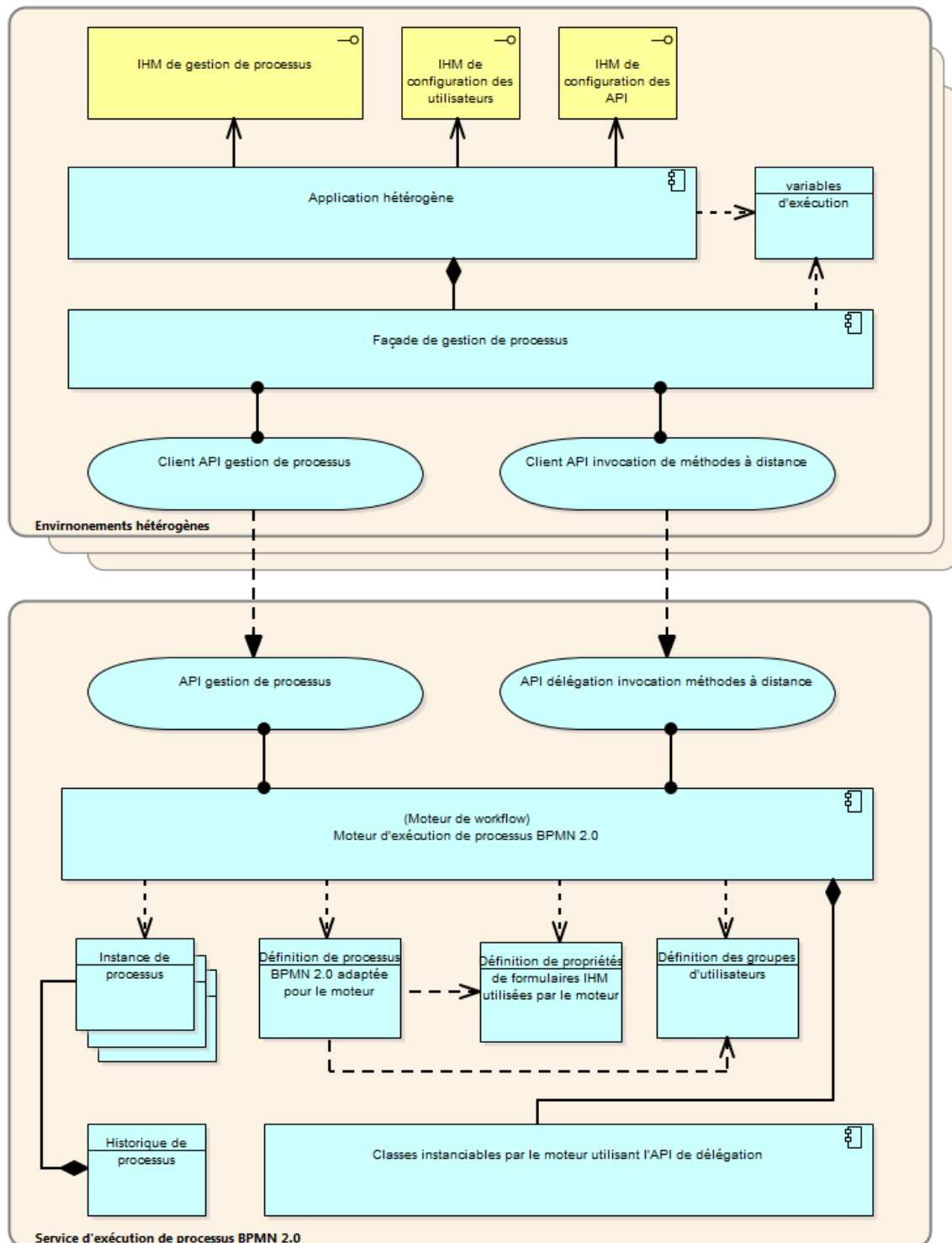


Figure 57 - Architecture distribuée d'exécution de processus d'entreprise sur des environnements techniquement hétérogènes.

VI Conclusion

Ce travail de mémoire m'a permis de comprendre comment il est possible de modéliser et d'exécuter des processus par le biais de différents outils de modélisation et de gestion de processus. Ces outils permettent de masquer complètement le système d'information de l'entreprise et de rendre accessible aux utilisateurs une interface logiciel unique dans laquelle ils peuvent retrouver les informations dont ils ont besoin au moment où ils en ont besoin.

Nous avons pu voir les difficultés liées à la modélisation de processus et de son instanciation dans des environnements technologiquement hétérogènes. L'objectif d'homogénéisation est atteint dès que nous optons pour un formalisme standardisé de notation de processus. L'objectif d'interopérabilité d'exécution de processus est étroitement dépendant du moteur de workflow utilisé et du contexte technologique dans lequel l'exécution de processus est faite. Ce mémoire a montré différentes solutions d'instanciation de processus et l'importance de placer le client au centre de ses préoccupations. L'utilisation d'une méthodologique générique et réutilisable est possible pour guider les différentes étapes de la transposition de processus métier afin d'aboutir à son implémentation et exécution dans des solutions technologiques hétérogènes. La genericité est une promesse d'économie de moyens et de réutilisabilité. L'application de cette méthodologie permet de se poser les bonnes interrogations quant à la modélisation et l'implémentation de processus.

Ayant l'habitude d'apporter du support technique à des projets informatiques d'intégration de besoins de commerce électroniques lié à des catalogues de produits, gestion de tarifications, flux de données dans les systèmes d'informations d'entreprise, architectures logiques et d'infrastructures, etc. J'ai pris beaucoup de plaisir en la réalisation de ce travail de mémoire portant sur des notions différentes liées au processus et globalement à l'automatisation. L'industrialisation de la production technique de solutions logicielles est intimement liée avec l'optimisation de processus par automatisation. La chaîne de production de logiciels informatique peut être améliorée en automatisant des processus humains répétitifs, fréquents et coûteux ce qui permet aux ingénieurs d'apporter une meilleure valeur ajoutée pour l'entreprise. Je pense essentiellement à l'utilisation d'outils permettant d'automatiser les tâches quotidiennes des développeurs implémentant des logiciels et des opérationnels ayant besoin de maintenir des infrastructures. Le travail de ce mémoire m'a permis d'avoir une vision plus haute sur l'amélioration de la production en entreprise.

Ma principale perspective est de pousser cette recherche plus loin par l'identification des moyens pour faire communiquer intelligemment les différents composants logiciels d'entreprise et l'intégration de moteurs de règles. Par exemple, les systèmes d'information en entreprise sont composés par des composants logiciels différents ayant besoin de communiquer. Cette communication par flux d'informations a besoin d'être intelligente par l'utilisation d'automates appliquant des règles métier de gestion. Nous pouvons résoudre cette problématique par l'utilisation de composants de processus automatiques dont les règles métiers sont décrites dans des définitions de processus exécutables.

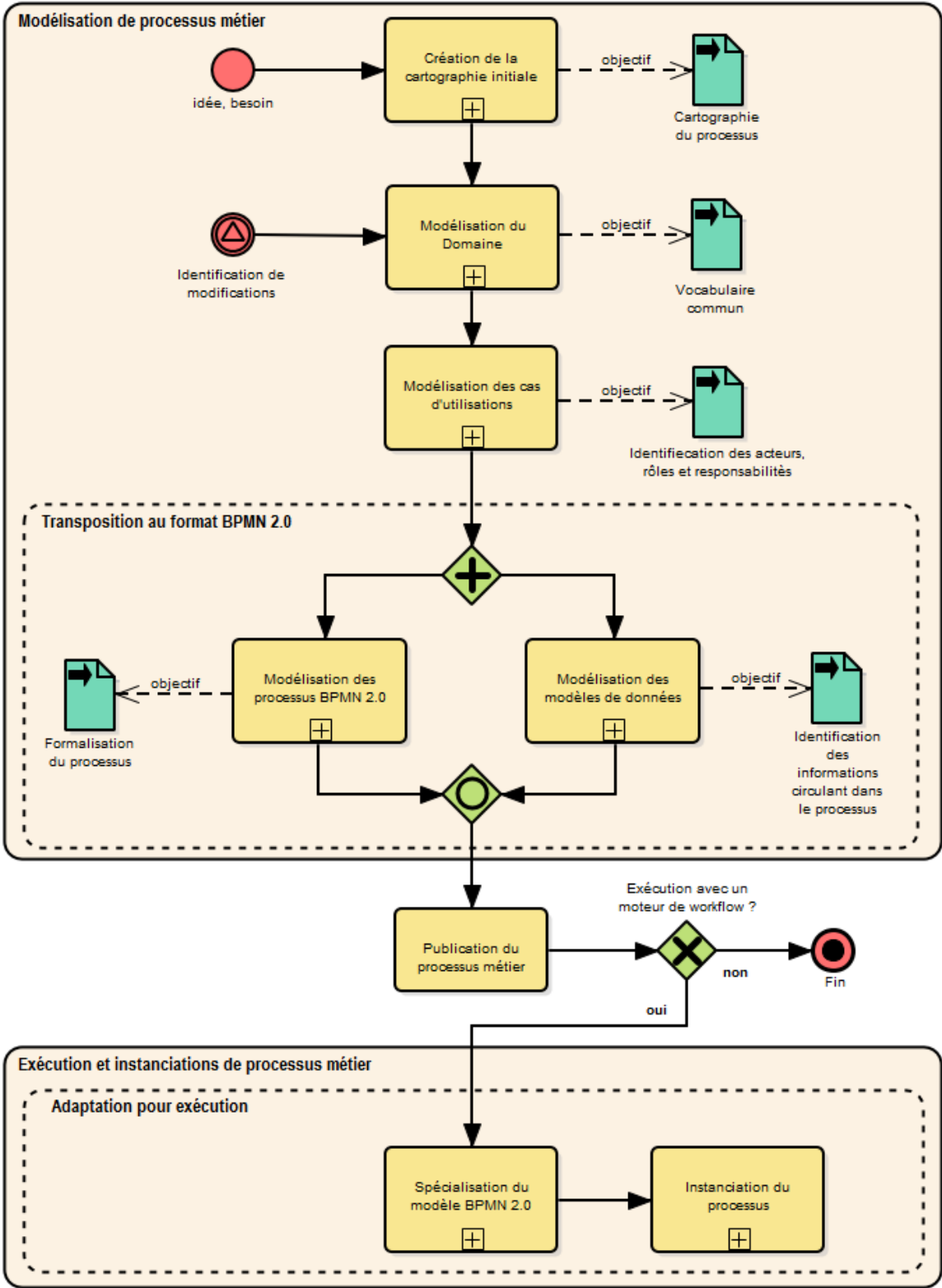
Bibliographie

1. **Wikipedia.** Espace de noms XML. *Wikipedia.* [En ligne] http://fr.wikipedia.org/wiki/Espace_de_noms_XML.
2. **Briol, Patrice.** *Ingenierie Des Processus Metiers, de L'Elaboration A L'Exploitation.* s.l. : Lulu.com, 2008. ISBN:9781409200406.
3. **Eriksson, H. and Penker, M.** *Business Modeling with UML: Business Patterns at Work.* s.l. : Wiley, 2000. ISBN-13: 978-0471295518.
4. **OMG.** Documents Associated With BPMN Version 2.0.2. *BPMN 2.0.* [Online] Janvier 20, 2013. <http://www.omg.org/spec/BPMN/2.0.2/>.
5. **Arlow, Jim.** Introduction To BPMN 2. *fr.slideshare.net.* [En ligne] 2009. <http://fr.slideshare.net/jimarlow/introductiontobpmn005>.
6. **Enterprise Architect.** Enterprise Architect - UML Design Tools and UML CASE tools for software development. *sparxsystems.com.au.* [Online] 2015. <http://www.sparxsystems.com.au/products/ea/index.html>.
7. **Chudley, Phil.** Working with BPMN 2.0 in Enterprise Architect version 11 / 12. *community.sparxsystems.com.* [En ligne] 12 01 2015. <http://community.sparxsystems.com/tutorials/833-working-with-bpmn-2-0-in-enterprise-architect-version-11-12>.
8. **Panagacos, Theodore.** *The Ultimate Guide to Business Process Management: Everything You Need to Know and How to Apply It to Your Organization.* s.l. : CreateSpace Independent Publishing Platform, 2012. isbn:9781477486139.
9. **Doug Rosenberg, Matt Stephens.** *Use Case Driven Object Modeling with UML Theory and Practice.* U.S.A : Apress, 2008. ISBN-13: 978-1-59059-774-3.
10. **Doug Rosenberg, Matt Stephens, and Mark Collins-Cope.** *Agile Development with ICONIX Process: People, Process, and Pragmatism.* s.l. : Apress, 2005. ISBN 1-59059-464-9.
11. **Activiti.** Activiti User Guide version 5.17.0. *activiti.org.* [En ligne] 2015. <http://www.activiti.org/userguide/>.
12. **Barrez, Tijs Rademakers & Tom Baeyens & Joram.** *Activiti in Action: Executable Business Processes in BPMN 2.0.* s.l. : Manning Publications, 2012. isbn:9781617290121.
13. **Potts, Jeff.** *Alfresco Developer Series, Working with Custom Content Types, 2nd Edition.* s.l. : ecmarchitect.com, 2012.
14. **Alfresco.** Content metamodel. *docs.alfresco.com.* [Online] Alfresco. <http://docs.alfresco.com/5.0/concepts/metadata-model-define.html>.
15. **Potts, Jeff.** *Alfresco Developer Series, Advanced Workflows, 2nd Edition.* s.l. : ecmarchitect.com, 2012.
16. **Farhdine Boutzakhti, Romain Felden.** DevOps, de l'intégration continue au déploiement continu. <http://www.ictjournal.ch/>. [En ligne] ictjournal, 01 09 2014. <http://www.ictjournal.ch/fr-CH/News/2014/12/02/DevOps-de-lintegration-continue-au-deploiement-continu.aspx>.
17. **Wikipedia.** Fichiers Java .properties. *Wikipedia.* [En ligne] <http://fr.wikipedia.org/wiki/.properties>.
18. **Kendall Scott, Doug Rosenberg.** *Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example.* s.l. : Addison-Wesley Professional, 2001. ISBN: 9780201730395.

Table des annexes

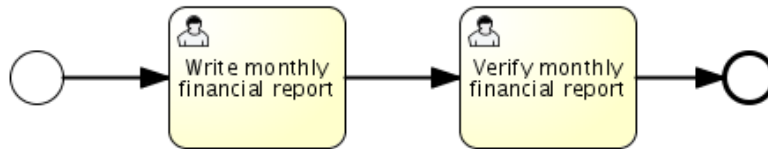
Annexe 1 : Etapes de la méthodologie appliquée – Synthèse	97
Annexe 2 : Exemple des deux niveaux de BPMN 2.0 (graphique et XML exécutable)	98
Annexe 3 : Cartographie initiale du processus NS-P1	99
Annexe 4 : Formalisation en BPMN 2.0 du processus NS-P1 dans Activiti.....	100
Annexe 5 : Captures d'écrans du processus NS-P1 dans Alfresco-Share	101
Annexe 6 : Eléments graphiques du standard BPMN 2.0	102

Annexe 1 : Etapes de la méthodologie appliquée – Synthèse



Annexe 2 : Exemple des deux niveaux de BPMN 2.0 (graphique et XML exécutable)

Représentation graphique BPMN 2.0 d'un processus de gestion de rapport financier mensuel (ou *financial report* en anglais) :



Définition XML-BPMN 2.0 de ce processus :

```
<definitions id="definitions"
  targetNamespace="http://activiti.org/bpmn20"
  xmlns:activiti="http://activiti.org/bpmn"
  xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL">

  <process id="financialReport" name="Monthly financial report reminder process">

    <startEvent id="theStart" />

    <sequenceFlow id='flow1' sourceRef='theStart' targetRef='writeReportTask' />

    <userTask id="writeReportTask" name="Write monthly financial report" >
      <documentation>
        Write monthly financial report for publication to shareholders.
      </documentation>
      <potentialOwner>
        <resourceAssignmentExpression>
          <formalExpression>accountancy</formalExpression>
        </resourceAssignmentExpression>
      </potentialOwner>
    </userTask>

    <sequenceFlow id='flow2' sourceRef='writeReportTask' targetRef='verifyReportTask' />

    <userTask id="verifyReportTask" name="Verify monthly financial report" >
      <documentation>
        Verify monthly financial report composed by the accountancy department.
        This financial report is going to be sent to all the company shareholders.
      </documentation>
      <potentialOwner>
        <resourceAssignmentExpression>
          <formalExpression>management</formalExpression>
        </resourceAssignmentExpression>
      </potentialOwner>
    </userTask>

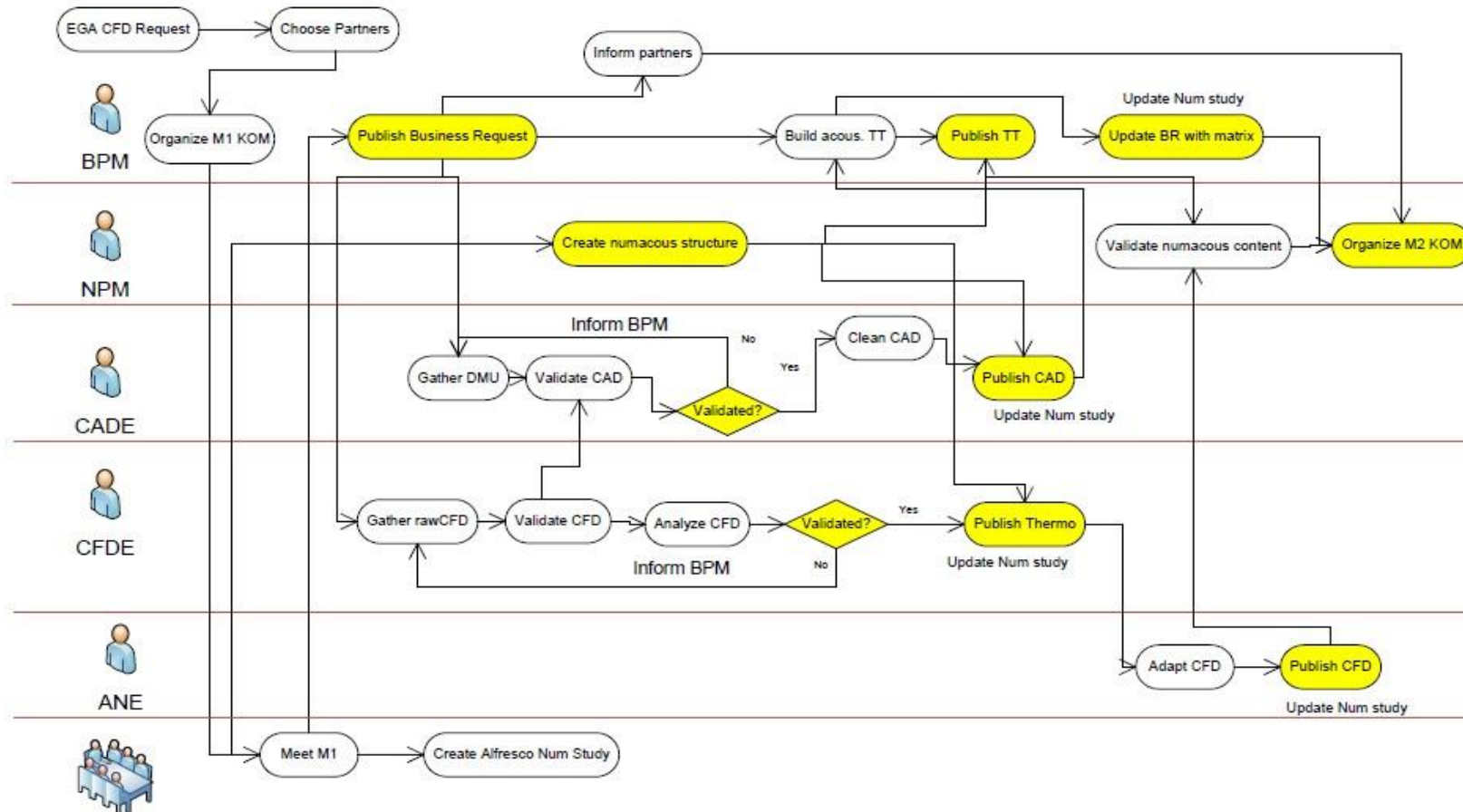
    <sequenceFlow id='flow3' sourceRef='verifyReportTask' targetRef='theEnd' />

    <endEvent id="theEnd" />

  </process>

</definitions>
```

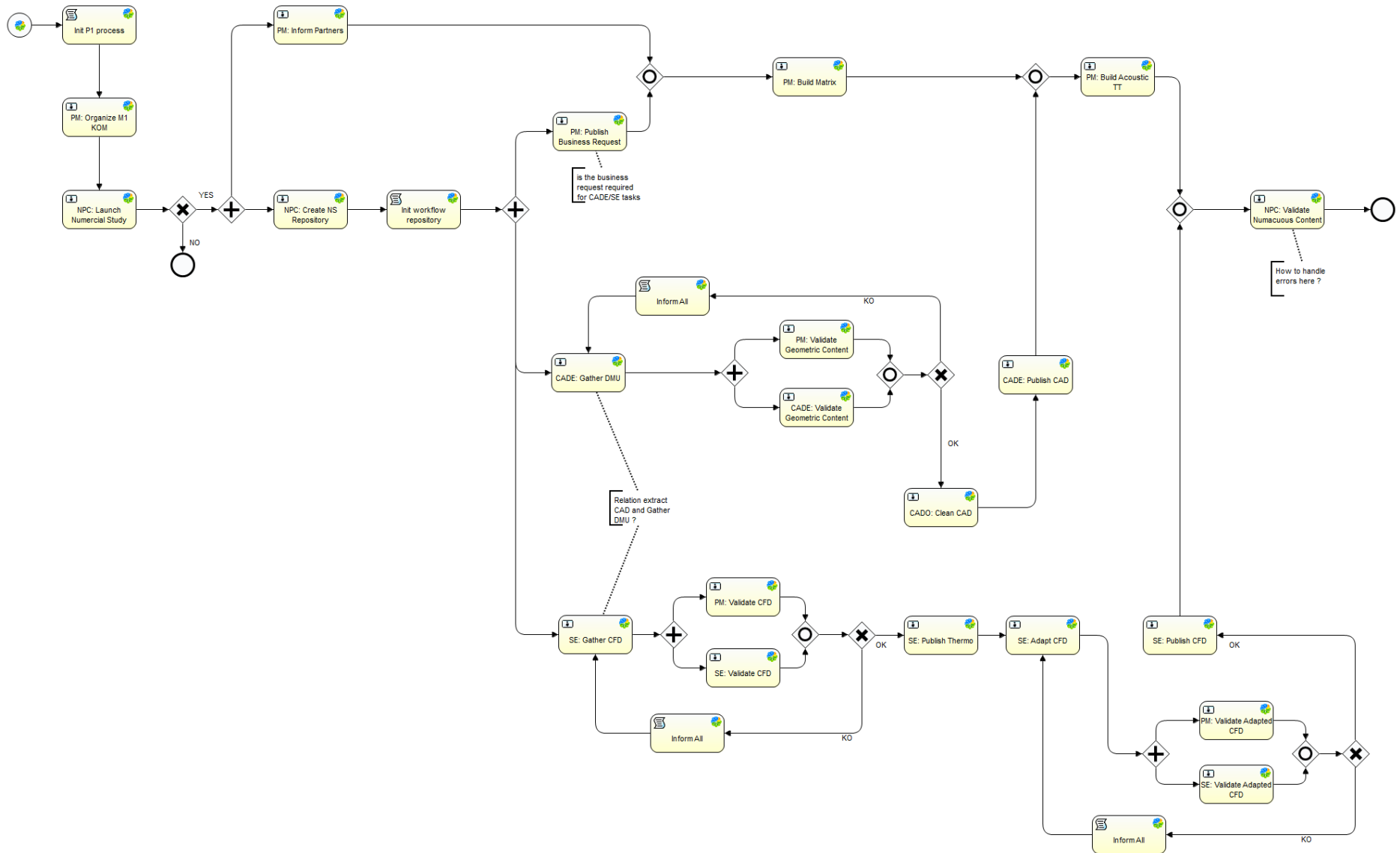
Annexe 3 : Cartographie initiale du processus NS-P1



- Bus. Project Manager (BPM)
- CFD Expert (CFDE)
- Numerical Project Manager (NPM)
- CAD Expert (CADE)
- Acous. Num. Expert (ANE)

Business workflow P1

Annexe 4 : Formalisation en BPMN 2.0 du processus NS-P1 dans Activiti



Annexe 5 : Captures d'écrans du processus NS-P1 dans Alfresco-Share



Alfresco logo

User Name
admin

Password

Login

© 2005-2012 Alfresco Software Inc. All rights reserved.

Active Workflows

- P1:The Study
Due: (None) Started: 10 June, 2013
Type: Numerical Study P1 Process (Activiti)
Description: Defines the P1 process for a new Numerical Study

Edit Task: NPC: Launch Numerical Study

Release to Pool

* Required Fields

Info

Numerical Study Name: The Study

Owner: Administrator Priority: Medium Due: (None)

Identifier: 47882

Assignee

P1 Role:
Numerical Process Coordinator

Progress

Status: *
Not Yet Started

Response

M1 Report:

Comment:

Approve Reject

Repository

- Repository
 - Acoustic Numerical Studies
 - [eefrtg] The Study
 - P1

Select

1 - 1 of 1 << 1 >>

(publish business request1)NOMPGM_inlet_flange_Business_request.docm (DEMANDE DE CAO / MAILLAGE)

Created just now by Administrator 40 KB

No Description

No Tags

Favorite Like 0 Comment Share

Annexe 6 : Eléments graphiques du standard BPMN 2.0

Activités

Tâche
Une tâche est le travail à être accompli, une unité de travail. La présence du symbole [] indique un sous-processus, une conversation décomposable.

Transaction
Une transaction est un ensemble d'activités placées sous le contrôle d'un protocole transactionnel.

Sous-processus événementiel
Un sous-processus événementiel est positionné à l'intérieur d'autres processus ou sous-processus. Il est activé lorsque son événement de début est déclenché. Il peut interrompre le processus de niveau supérieur ou rouler en parallèle (sans interruption), en fonction de l'événement de début.

Activité appelante
Une activité appelante englobe une tâche ou un sous-processus défini globalement et réutilisé dans ce processus.

Marqueurs d'activités
Les marqueurs indiquent le comportement des activités lors de leur exécution:

- Sous-processus
- Boucle
- Instances multiples en parallèle
- Instances multiples séquentielles
- Ad Hoc
- Compensation

Types de tâches
Le type spécifie la nature de l'action à être réalisée:

- Envoi
- Reception
- Usager
- Manuelle
- Regle d'affaires
- Service
- Script

Flux de séquence
détermine l'ordre d'exécution des activités.

Flux de défaut
est l'embranchement pris lorsque toutes les autres conditions sont évaluées comme fausses.

Flux conditionnel
à une condition assignée, qui détermine si ce flux doit ou ne doit pas être utilisé.

Conversations

Une communication définit un ensemble d'échange de messages logiquement associés. La présence du symbole [] indique une sous-conversation, une conversation décomposable.

Un lien de conversation relie une communication et un participant.

Un lien de conversation fourchu relie une communication et des participants multiples.

Diagramme de conversation

Chorégraphies

Une tâche de chorégraphie représente une interaction (échange de messages) entre deux participants.

Le marqueur « multiples participants » indique la présence de multiples participants du même type.

Un sous-processus de chorégraphie contient les détails d'une chorégraphie avec ses interactions.

Diagramme de chorégraphie

Événements

Aucun: indiquent généralement un déclenchement, un changement d'état ou la fin d'un processus.

Message: réception et envoi de messages.

Minuterie: cycle temporel, moment déterminé ou délai écoulé.

Escalade: mesure d'escalade à un niveau supérieur de responsabilité.

Conditionnel: réaction à un changement de condition ou à une règle d'affaires.

Lien: liaison de pagination. Deux événements correspondants équivalent à un flux de séquence.

Erreur: réception ou émission d'erreurs prévues.

Annulation: réaction à l'annulation d'une transaction ou déclenchement d'une annulation.

Compensation: gestion ou déclenchement d'une compensation.

Signal: signalisation entre différents processus. Un signal émis peut être capté plusieurs fois.

Multiple: réception d'un des événements spécifiques. Émission de tous les événements spécifiques.

Multiple parallèle: réception de tous les événements se produisant en parallèle.

Arrêt: déclenchement de la fin immédiate du processus.

	Niveau supérieur	Debut	Intermédiaire	Fin
Aucun	○	○	○	○
Message	✉	✉	✉	✉
Minuterie	⌚	⌚	⌚	⌚
Escalade	⬆	⬆	⬆	⬆
Conditionnel	⚖	⚖	⚖	⚖
Lien	➡	➡	➡	➡
Erreur	⚠	⚠	⚠	⚠
Annulation	⌛	⌛	⌛	⌛
Compensation	⏪	⏪	⏪	⏪
Signal	📡	📡	📡	📡
Multiple	⊕	⊕	⊕	⊕
Multiple parallèle	⊕	⊕	⊕	⊕
Arrêt	⦿	⦿	⦿	⦿

Traduction française: **Trisotech**

Diagramme de collaboration

Branchements

Branchement exclusif
Lors d'une division, le flux de séquence est passé exclusivement à un embranchement. Lors d'une convergence, il attend l'arrivée d'un seul flux de séquence entrant pour déclencher le flux de séquence sortant.

Branchement exclusif - basé sur événement
Est toujours suivi d'événements ou de tâches de type réception. Le flux de séquence est passé à l'événement/tâche qui se produit en premier.

Branchement parallèle
Pour la division d'un flux de séquence, tous les embranchements sont activés simultanément. Lors d'une convergence de flux parallèles, il attend que tous les flux entrants soient complétés pour déclencher le flux sortant.

Branchement inclusif
Lors d'une division, un ou plusieurs embranchements sont activés. Tous les embranchements doivent être complétés avant de finaliser une convergence.

Branchement exclusif - basé sur événement (début)
Chaque arrivée d'un événement démarre une nouvelle instance du processus.

Branchement complexe
Une division ou une convergence complexe est utilisée lorsque son comportement ne peut être capturé par les autres types de branchement.

Branchement parallèle - basé sur événement (début)
L'arrivée de tous les événements démarre une nouvelle instance du processus.

Pistes et corridors

Une piste identifie un participant au processus. Un corridor est une subdivision d'une piste ou d'un processus. Les corridors peuvent être attachés à des pistes, des activités ou à d'autres corridors.

Un flux de message indique le flux d'information entre les frontières organisationnelles. Les flux de message peuvent être présents en combinant des flux de message et des flux de séquence.

L'ordre des échanges de messages peut être présenté en combinant des flux de message et des flux de séquence.

Données

Une donnée d'entrée est une source externe au processus. Elle peut être lue par une activité.

Une donnée de sortie est un paramètre, cette variable est disponible à l'ensemble du processus.

Un objet de données représente une structure d'information généralement traitée dans les activités comme les documents, courriels, lettres, etc.

Une collection d'objets de données représente un ensemble d'information, telle une liste de articles d'une commande.

Un dépôt de données est un endroit où le processus peut lire et écrire des données, comme une base de données ou un classeur. Son contenu persiste même après la fin de l'instance du processus.

Un Message est utilisé pour représenter le contenu d'une communication entre deux participants.

Liste des figures

Figure 1 - Représentation d'un processus métier selon le concept d'Eriksson-Penker.....	11
Figure 2 - Exemple d'un processus simple en notation Eriksson-Penker.....	12
Figure 3 - Etapes de transposition d'un processus métier - focus modélisation.	15
Figure 4 - Historique de l'évolution du standard BPMN.	17
Figure 5 - Eléments de base BPMN 2.0.	17
Figure 6 - Exemple d'un processus simple en notation BPMN 2.0.....	18
Figure 7 - Carté heuristique du cœur de la spécification BPMN 2.0.	19
Figure 8 – Ecran de modélisation de processus avec le logiciel web Activiti Modeler.	21
Figure 9 – Ecran de modélisation UML avec le logiciel Enterprise Architect 12.....	22
Figure 10 - Roue de Deming (PDCA).....	25
Figure 11 - Cycle BPM.....	25
Figure 12 - Relation étapes PDCA / Cycle BPM.....	25
Figure 13 - Diagramme de domaine du processus NS-P1.	29
Figure 14 - Processus NS-P1 cas d'utilisation de l'acteur NPC.....	30
Figure 15 - Symboles de diagramme de <i>robustness</i> ICONIX.....	31
Figure 16 - Synthèse de la méthodologie <i>ICONIX</i>	32
Figure 17 - Liste des rôles du processus NS-P1 sous la forme d'acteurs UML.....	34
Figure 18 - Regroupement des cas d'utilisation par acteur du processus NS-P1.....	35
Figure 19 - Modèle de données du processus NS-P1.....	36
Figure 20 - Types de <i>Task</i> BPMN 2.0 principalement utilisés.	38
Figure 21 - Types des branchements BPMN 2.0 principalement utilisés.	40
Figure 22 - Pattern BPMN 2.0 de boucles.....	41
Figure 23 - Pattern BPMN 2.0 d'exécution parallèle.	41
Figure 24 - Patterns BPMN 2.0 d'interruption.	42
Figure 25 - Types d'événements BPMN 2.0 principalement utilisés.....	42
Figure 26 - Etapes de transposition d'un processus métier, focus instanciation et exécution.	44
Figure 27 - Etapes d'une étude numérique avionique.	48
Figure 28 - Vue d'ensemble du processus NS, en anglais, des parties principales.....	49
Figure 29 - Processus NS en notation Eriksson-Penker.....	50
Figure 30 - Méthodologie projet informatique X2i : le cycle itératif « agile ».	53
Figure 31 - Vue d'ensembles des outils Activiti.....	56
Figure 32 - Ecran de l'espace de travail du plug-in Eclipse : Activiti Designer.	57
Figure 33 - Ecrans de l'application web Activiti Explorer.....	58
Figure 34 - Processus BPMN 2.0 montrant des IHM de tâches d'interactions humaines extraites de l'outil Activiti Explorer.	59
Figure 35 - Mise en avant des catégories (ou niveaux) BPMN 2.0 utilisées par Activiti.....	61
Figure 36 - Projection des Acteurs sur Activiti et Alfresco.	62
Figure 37 - Configuration de gestion de formulaires par délégation dans Activiti Designer.	64
Figure 38 - Formulaire <i>Share Form</i> d'un <i>User Task</i> du processus NS-P1.....	65
Figure 39 - Modèle de données d'un processus Activiti.	66
Figure 40 - Exemple 1 de prédicat sur un branchement exclusif.	68
Figure 41- Exemple 2 de prédicat sur un branchement exclusif.	68
Figure 42 - Procédé d'exécution de processus BPMN 2.0 sur Alfresco.	70
Figure 43 - Méta modèle du modèle de contenu sous Alfresco (14).....	71
Figure 44 - Définition du <i>Task Model</i> Alfresco du processus NS-P1.	72
Figure 45 - Projection des attachements documentaires du processus NS-P1 sur l'entrepôt de documents Alfresco.	74
Figure 46 - Configuration de l'espace de travail dans Activiti Designer.	76
Figure 47 - Architecture en 3 couches de l'implémentation d'un processus avec Activiti et Alfresco.	79
Figure 48 - Exemple d'interopérabilité avec des solutions techniques hétérogènes.	81
Figure 49 - Modélisation du domaine des circuits de validations.	83

Figure 50 - Diagramme de cas d'utilisations des circuits de validations.	84
Figure 51 - Cartographie initiale des circuits de validations.....	85
Figure 52 - Modélisation au format BPMN 2.0 du processus de circuit de validation.....	85
Figure 53 - Diagramme de domaine d'un moteur d'exécution de processus BPMN 2.0.	87
Figure 54 - Diagramme de classes UML du moteur de workflow circuit de validation	89
Figure 55 - IHM de définition d'un circuit de validation	90
Figure 56 - IHM de formulaire de validation d'une demande d'achat avec issue.	91
Figure 57 - Architecture distribué d'exécution de processus d'entreprise sur des environnements techniquement hétérogènes.....	93

Index

- Activiti, 14, 19, 20, 21, 26, 42, 47, 51, 52, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 74, 75, 76, 77, 78, 79, 81, 86, 88
- adaptation, 26, 44, 56, 60, 69, 72, 73
- Alfresco, 14, 48, 51, 52, 54, 55, 56, 57, 58, 60, 61, 62, 63, 64, 65, 67, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 86, 90, 101, 106
- API, 55, 63, 67, 81, 86, 88
- Archimate 2, 22, 78, 93
- B2B, 3, 10, 82, 88, 91
- Bonitasoft, 55, 60, 81
- BPM, 23, 24, 37, 55, 77, 82
- BPML, 5, 16
- BPMN 2.0, 3, 4, 5, 10, 13, 15, 16, 17, 18, 19, 20, 21, 22, 25, 26, 36, 37, 38, 39, 40, 41, 42, 43, 45, 46, 47, 48, 51, 52, 54, 55, 56, 57, 59, 60, 61, 62, 67, 68, 69, 70, 71, 77, 81, 85, 86, 87, 88, 89, 98, 100, 102, 106
- branchement, 17, 39, 40, 41, 60, 67, 68
- Business Archive, 76
- cartographie, 12, 19, 32, 33, 34, 35, 36, 37, 38, 39, 54, 84, 85
- cas d'utilisations, 29, 30, 31, 35, 84
- circuits de validation, 82, 83, 86
- définition de processus, 5, 10, 13, 14, 15, 18, 42, 44, 45, 46, 55, 56, 57, 60, 62, 64, 67, 70, 71, 72, 73, 75, 76, 77, 80, 81, 86, 87, 88
- deployment bean, 70
- domaine, 28, 29, 30, 33, 35, 36, 39, 42, 50, 66, 83, 86, 87
- End Event, 42
- Enterprise Architect, 20, 22, 26, 28, 36, 37, 54
- Eriksson-Penker, 11, 12, 22, 36, 50
- Gateway, 3, 39, 40, 41
- génie logicielle, 23
- ICONIX, 23, 27, 29, 31, 32, 34, 35, 42, 84, 106
- IHM, 3, 4, 19, 26, 28, 30, 39, 45, 58, 59, 61, 63, 65, 69, 71, 74, 75, 76, 77, 80, 86, 90, 91
- instanciation de processus, 10, 13, 16, 56, 81
- interopérabilité, 14, 16, 46, 81, 106
- jbPM, 55
- Magento, 28, 82, 83, 86, 88, 89, 90, 106
- Message Task, 39
- modèle de données, 4, 35, 36, 65, 66, 71, 75
- modélisation de processus, 13, 15, 21, 37, 48, 83
- moteur d'exécution, 20, 27, 45, 55, 56, 57, 76, 86, 88, 106
- moteur de workflow, 6, 18, 19, 20, 32, 38, 39, 42, 43, 44, 45, 46, 47, 51, 52, 54, 55, 56, 57, 60, 61, 63, 66, 67, 68, 69, 74, 77, 78, 81, 86, 88, 89
- OMG, 3, 4, 5, 16, 18, 46, 47, 61, 78
- processus métier, 9, 10, 11, 13, 14, 15, 16, 19, 24, 26, 32, 35, 37, 38, 44, 45, 48, 49, 55, 56, 59, 60, 73, 80, 81
- projet *Acoustic*, 16, 18, 20, 23, 28, 30, 33, 34, 36, 40, 43, 45, 47, 55, 56, 58, 60, 63, 64, 65, 69, 73, 76, 82, 83, 92
- projet Descours & Cabaud, 23, 33, 56, 58, 82
- robustness, 30, 31
- Script Task, 39, 67, 68
- Sequence Flow, 3, 68
- Sequence Flows*, 17, 67, 68
- Service Task, 39, 67, 68
- Share Forms, 58, 63, 64, 65, 74, 78
- Start Event, 42, 60, 63, 66, 67
- tests unitaires, 32, 55, 57, 58, 73, 89, 91
- transposition, 15, 37, 42, 44
- UML, 3, 4, 5, 11, 20, 22, 26, 27, 28, 29, 30, 31, 33, 34, 36, 54, 55
- User Task, 4, 39, 42, 59, 61, 62, 63, 65, 66, 67, 71, 72, 74, 77
- variables d'exécution, 36, 45, 57, 60, 67, 88
- variables de processus, 36, 59, 60, 66, 72, 86, 87
- X2i, 2, 9, 20, 28, 48, 50, 52, 53, 70, 82, 83
- XML, 3, 4, 5, 16, 18, 19, 20, 39, 40, 43, 45, 46, 47, 55, 57, 58, 60, 62, 63, 64, 67, 68, 69, 76, 81, 87, 88, 98

MEMOIRE D'INGENIEUR C.N.A.M. en INFORMATIQUE

Modélisation et instanciation de processus sur des solutions techniquement hétérogènes

Thierry Vercausse

Toulouse, le 2 juillet 2015

RESUME

L'informatisation des processus métiers en entreprise est une étape essentielle dans l'industrialisation. Cette industrialisation permet aux entreprises de gagner du temps et elles peuvent se tourner vers des domaines à plus forte valeur ajoutée. La modélisation est une étape fondamentale dans l'informatisation de processus puisqu'elle permet d'étudier et de rationaliser les processus d'entreprises. La spécification BPMN 2.0 est un moyen de modéliser un processus avec des diagrammes et une syntaxe évoluée permettant son exécution. Les moteurs de workflows sont des outils de gestion de processus qui ont la vocation d'instancier et d'exécuter des processus et peuvent dans des architectures orienté services être utilisés pour la gestion de processus avec des logiciels techniquement hétérogènes. Un travail de modélisation et d'exécution de processus est réalisé avec l'application d'une méthodologie commune sur deux cas d'études ayant un environnement technologique et un contexte projet différents. Ce travail permet de se rendre compte des potentiels, des difficultés et des problèmes auxquels il faudra apporter des solutions pour rendre l'implémentation de processus sur des environnements techniquement hétérogènes possibles.

Mots clés : Processus métiers, modélisation de processus, BPMN 2.0, ICONIX, moteur d'exécution de processus, interopérabilité, Alfresco, Activiti, Magento.

SUMMARY

Computerization of business processes is an essential step towards industrialization. This allows companies to save time and to focus on higher value-added activities. Modeling business processes is a fundamental step toward the computerization of business processes since it allows the study and streamlining of processes. The BPMN 2.0 specification is a way to model business processes using diagrams and an evolved syntax that allows process execution. Workflow engines are business process management tools having the vocation to instantiate and execute business processes which can be used in service oriented architectures to manage business processes in heterogeneous application software environments. A work of process modeling and execution is done using a common methodology on two case studies having different technological environments and project contexts. This work measures the possibilities, difficulties and problems for which solutions are required to be able to implement processes in technically heterogeneous environments.

Keywords: Business process, process modeling, BPMN 2.0, ICONIX, workflow engine, interoperability, Alfresco, Activiti, Magento.