

# RÉALISATION D'UN MODULE DE TRADUCTION DE DONNÉES



Cyrille PIERRE  
IUT de Clermont-Ferrand - Département informatique.  
Stage de 10 semaines dans la société Compilsoft.  
Année 2011-2012



## REMERCIEMENTS

Je tiens d'abord à remercier l'entreprise Compilsoft de m'avoir permis de réaliser mon stage dans leur entreprise. Je souhaite également remercier :

- Yves PETIT (maître de stage) pour m'avoir guidé et suivi mon travail durant tout le stage,
- Mickael SORIANO pour m'avoir apporté toutes les informations nécessaires pour pouvoir mettre en place l'application,
- Gautier BOIS pour son écoute et ses encouragements,
- Jérémy DENIS (stagiaire de l'IUT) pour avoir travaillé avec moi et pour m'avoir aidé lorsque je rencontrais des difficultés,
- Thomas LEIRIS pour ses nombreux conseils concernant les outils et les langages utilisés lors de mon stage,
- Païkan CONSTANCEAU ainsi que tous les membres du pôle PAO pour m'avoir accueilli parmi eux.



## SOMMAIRE

Introduction .....	4
1. Présentation de Compilsoft .....	5
1.1. La structure de l'entreprise .....	5
1.1.1. Description de la société .....	5
1.1.2. Organisation des différents pôles .....	5
1.2. Les services .....	6
1.3. Les logiciels .....	7
1.3.1. CS MediaContent .....	7
1.3.2. CS AutoPage .....	8
2. Identification du travail à réaliser .....	9
2.1. Contexte du projet .....	9
2.2. Analyse de l'application existante .....	9
2.2.1. Description du fonctionnement de l'E-Translate .....	9
2.2.2. Structure de l'application .....	12
2.2.3. Description de la base de donnée .....	14
2.3. Description des outils de travail .....	16
2.3.1. Les logiciels .....	16
2.3.2. Le framework PRADO .....	17
2.4. La plateforme Activiti .....	17
3. Réalisation de l'application .....	18
3.1. Le cahier des charges .....	18
3.1.1. Les nouveautés apportées .....	18
3.1.2. Les éléments de l'application .....	20
3.1.3. Le workflow .....	23
3.2. Analyse de l'application .....	25
3.2.1. Déploiement de l'application auprès du MediaContent .....	25
3.2.2. La base de donnée .....	26
3.2.3. La structure de l'application .....	29
3.3. La méthode Scrum .....	30
3.3.1. L'organisation d'un projet .....	30
3.3.2. L'organisation de l'équipe .....	31
3.3.3. Le fonctionnement de Scrum .....	32
3.4. Détail de la programmation .....	33
3.4.1. Le premier sprint .....	33
3.4.2. Le deuxième sprint .....	35
3.4.3. Le troisième sprint .....	37
Bilan technique .....	39
Conclusion .....	40
Summary .....	41
Bibliographie .....	42
Lexique .....	43



## INTRODUCTION

Dans le cadre de ma deuxième année à l'Institut Universitaire de Technologie de Clermont-Ferrand, j'ai effectué un stage de 10 semaines dans la société Compilsoft.

L'entreprise Compilsoft possède actuellement une application web appelée MediaContent qui permet de gérer du contenu pour d'autres entreprises. Cette application possède également des modules permettant d'effectuer différentes actions sur ce contenu.

Le sujet du stage va concerner un module du MediaContent nommé E-Translate qui s'occupe de gérer la traduction des données dans différentes langues. L'entreprise possédait déjà une application E-Translate mais elle ne convenait pas à l'équipe de développement. C'est pourquoi mon stage consistera à la réécriture complète de cette application.

L'équipe de développement chargée de la réécriture de cette application était composée de deux stagiaires provenant tous les deux de l'IUT de Clermont-Ferrand. L'autre stagiaire qui a travaillé avec moi est Jérémy DENIS.

Ce rapport va donc expliquer en détail les étapes qui ont été nécessaires pour créer cette application. L'explication commencera par l'analyse de l'ancienne application ainsi que les outils utilisés par l'entreprise. Elle enchaînera ensuite par la conception du cahier des charges de la nouvelle application ainsi qu'une explication plus technique sur son fonctionnement. Mais avant cela, le rapport commencera par une présentation de l'entreprise Compilsoft.



## 1. Présentation de Compilsoft

### 1.1. La structure de l'entreprise

#### 1.1.1. Description de la société

Fondée en 2006, Compilsoft est une société offrant des services et des logiciels pour des entreprises clientes en utilisant le concept SaaS (Software as a Service). Cela signifie qu'elle propose un abonnement à un logiciel plutôt qu'une licence.

L'un des principaux logiciel en ligne que propose cette société est la gestion de l'information Produit avec CS MediaContent. Elle est également leader en France en ce qui concerne la publication multicanal de catalogues grâce à CS AutoPage.

Compilsoft réunit deux secteurs d'activité : la chaîne graphique et l'informatique. Elle permet ainsi de pouvoir gérer les données d'une entreprises jusqu'à la modélisation d'un catalogue.

#### 1.1.2. Organisation des différents pôles

L'entreprise possède deux établissements différents. Celui se trouvant dans la ville de LUNEL correspond au pôle commerciale et celui de BÉDARIEUX correspond au pôle technique.

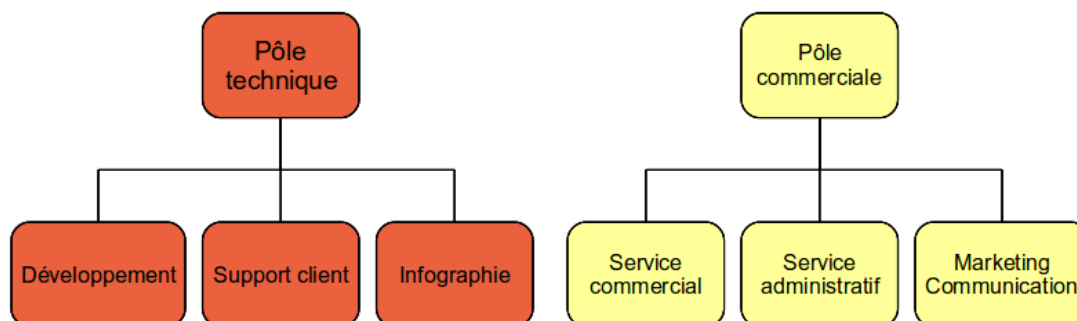


Fig. 1.1 : Organigramme de la société Compilsoft

Le pôle technique est composé de 3 services différents :

- Le service de développement s'occupe de la programmation et la maintenance des applications web fournies aux clients. Il est constitué d'une équipe de 4 développeurs (Mikaël SORIANO, Yves PETIT, Thomas LEIRIS et Païkan CONSTANCEAUX) et est encadré par le directeur technique (Gauthier BOIS).
- Le service de support client permet aux utilisateurs des logiciels développés par Compilsoft de pouvoir rapporter à l'entreprise les modifications qu'ils souhaitent avoir et également les bugs rencontrés. Les personnes qui s'occupent de ce service sont aussi le directeur techniques et les développeurs.



- Le service d'infographie correspond à la création graphique de catalogues à partir des données des entreprises clientes. L'équipe d'infographiste est composée de 3 personnes.

Le pôle commerciale est lui aussi composé de 3 services :

- Le service commercial regroupe les personnes communiquant directement avec les entreprises clientes afin de décider les logiciels et les services que Compilsoft pourra leur fournir.
- Le service administratif dont le fondateur, Pierre CHOTY fait partie, représente la direction de la société.
- Le service marketing et communication s'occupe quant à lui, de faire connaître la société dans toute la France mais également dans l'Europe.

### 1.2. Les services

Compilsoft propose aux entreprises clientes toute une gamme de services informatiques dans le domaine de la gestion des données :

- **Formation** : Compilsoft propose à ses clients de courtes formations sur l'utilisation des différents outils créés par la société. Elle s'adresse aux administrateurs et aux développeurs qui devront gérer les produits de façon autonome.
- **Expertise graphique** : Les documents et les publications des entreprises clientes sont réalisés par l'équipe graphique de Compilsoft.
- **Audit et analyse** : Ce service a pour but d'optimiser la production des documents grâce à l'expérience gagnée auprès des clients. Il consiste à s'occuper de la gestion des données, de la planification et de la maîtrise d'œuvre.
- **Support et assistance** : Grâce au Contrat Crédit Pack, les clients peuvent disposer d'un crédit temps qu'ils pourront utiliser en fonction de leurs besoins.
- **Développement** : Compilsoft dispose d'une équipe technique accompagnant les clients dans leurs projets afin de gérer au mieux leurs informations.

### 1.3. Les logiciels

Les logiciels développés par Compilsoft utilisent 2 plate-formes de travail différentes. La majeure partie des applications disponibles sont des applications Web, mais certaines applications sont intégrées sous forme de modules dans des applications externes comme Quark XPress ou Adobe InDesign.

Voici un diagramme représentant les deux principaux logiciels de Compilsoft :

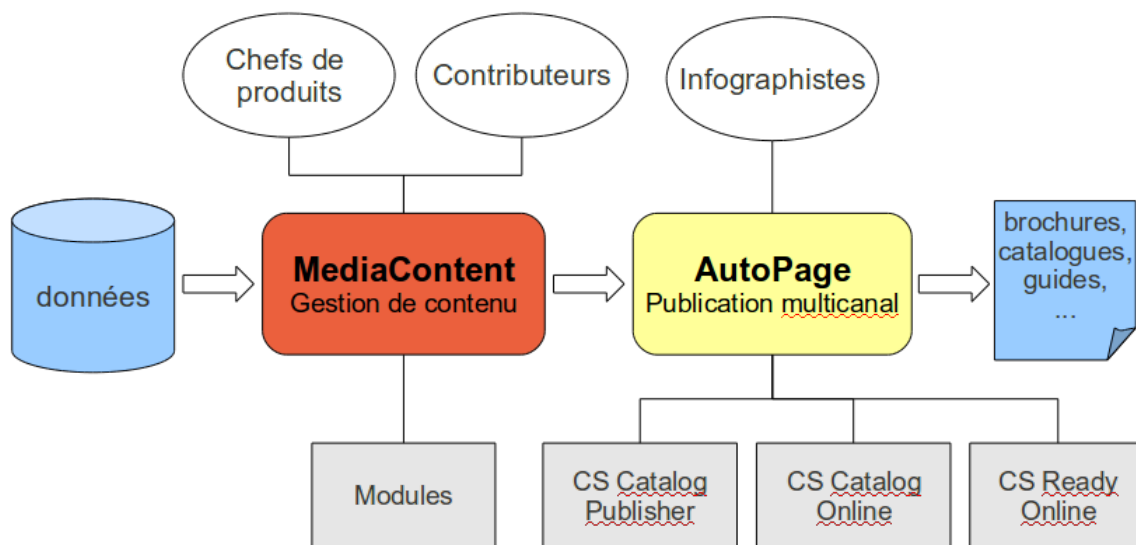


Fig. 1.2 : Représentation du fonctionnement des outils de Compilsoft

L'entreprise cliente va commencer par importer ses données dans le MediaContent afin que les contributeurs puissent y effectuer tous les traitements qu'ils souhaitent. Une fois que les données seront prêtes, les infographistes pourront modéliser les documents à partir des applications de la plate-forme AutoPage. Ces documents pourront être numériques ou imprimés.

#### 1.3.1. CS MediaContent

Le MediaContent est une application de gestion de contenu. Il intègre divers outils sous la forme de modules, permettant d'effectuer des traitements sur une base de données contenant les informations de l'entreprise cliente. Cette plate-forme est donc destinée aux contributeurs et aux chefs de produits qui souhaitent pouvoir stocker, centraliser et organiser l'information.

Il possède beaucoup d'outils dont la principale utilité tourne autour de la gestion d'informations personnelles (PIM). Il permet, entre autres, de pouvoir gérer les données sous forme d'arborescences et également de pouvoir y intégrer des images et des attributs (descriptifs, prix, références...). Le MediaContent étant une plate-forme collaborative, il intègre également des outils de travail collaboratif, de suivi de production et de validation des documents, sans oublier la gestion des droits d'accès des utilisateurs. Ces données pourront être utilisées par la suite afin de créer des modélisations graphiques. C'est pourquoi le MediaContent intègre : un outil d'importation, d'exportation et de synchronisation avec la plupart des bases de données.



### 1.3.2. CS AutoPage

AutoPage est le nom de la plate-forme utilisée par les infographistes pour pouvoir modéliser des documents ou créer des web services à partir d'informations stockées dans le MediaContent. Le contenu de sortie pourra être multimédia comme par exemple un catalogues interactif ou un service d'e-commerce, mais il pourra également s'agir de documents papiers comme des brochures, des annuaires, des guides...

AutoPage peut être découpée en trois solutions de publication :

- **CS Catalog Publisher** : Cette solution permet de mettre en page automatiquement des catalogues ou des documents marketing. Sa particularité est qu'elle utilise des logiciels externes comme Quark XPress ou Adobe InDesign simplifiant ainsi son utilisation par des utilisateurs déjà habitués à ces outils. Les données peuvent être importées à partir de fichier Excel et Xml, ou en utilisant une base de donnée.
- **CS Catalog Online** : Cette solution est utilisée pour la publication dynamique et interactive sur Internet. Elle offre la possibilité de publier du texte, des images et des vidéos sans avoir spécialement des connaissances informatiques. Elle permet ainsi d'accroître la visibilité de l'entreprise cliente.
- **CS Ready Online** : Cette dernière solution permet de mettre à disposition tout type de documents personnalisables sans connaissance graphique. Ces documents pourront ainsi être accessibles simplement avec un navigateur. Elle est donc destinée à faciliter la gestion et l'animation des espaces de commercialisation comme les magasins, les filiales, les agences...





## 2. Identification du travail à réaliser

### 2.1. Contexte du projet

Le stage s'est déroulé à BÉDARIEUX dans l'équipe de développement du pôle technique. Actuellement, cette équipe travaille sur l'application MediaContent et les modules qui lui sont liés. Ils s'occupent de régler les problèmes et les bugs encore présents dans ces applications chez certains clients. Ils essayent également d'améliorer le fonctionnement des différents modules en y intégrant de nouvelles fonctionnalités.

Parmi ces modules on peut par exemple citer :

- Print Publishing qui permet de gérer la publication des données,
- E-Sourcing qui est un module de relations fournisseurs,
- E-Translate qui est un module de gestion de la traduction des données.

Le sujet de mon stage, y compris celui de Jérémy DENIS (l'autre stagiaire provenant également de l'IUT de Clermont-Ferrand) va consister à recréer à partir de zéro le module E-Translate.

Le module E-Translate actuel ne convient pas aux clients. Certains lui reprochent de ne pas être assez ergonomique. De plus l'équipe de développement souhaiterait y voir apparaître de nouvelles fonctionnalités permettant ainsi aux clients de pouvoir travailler de façon plus efficace. Cette refonte complète permettrait aussi de pouvoir y intégrer un nouvel outil nommé Activiti qui s'occuperait de la gestion des activités d'une entreprise.

### 2.2. Analyse de l'application existante

Avant de s'intéresser à la création du nouveau module E-Translate, il a fallu d'abord comprendre comment fonctionnait celui existant afin de pouvoir utiliser la même méthode de programmation. Il a fallu cependant ne pas reprendre exactement les mêmes interfaces afin de ne pas se retrouver avec les mêmes problèmes que l'ancienne version d'E-Translate.

#### 2.2.1. Description du fonctionnement de l'E-Translate

L'application MediaContent ainsi que ses modules sont des applications accessibles via un navigateur. Par conséquent, ils sont écrits à partir de langages de programmation spécialisés pour les applications web. L'équipe de développement a choisi d'utiliser le langage PHP pour programmer le module E-Translate contrairement au MediaContent qui lui utilise le Java.

Ce module est utilisé directement par les employés des entreprises clientes ou par des personnes externes à l'entreprise afin de pouvoir gérer la traduction des informations contenues dans la base de données du MediaContent. Ces informations seront stockées dans E-Translate sous forme de fiches à remplir par les utilisateurs qui auront pour rôle de traduire, relire ou valider ces fiches.



Actuellement l'application E-Translate est composée de deux parties :

- la partie administration dans laquelle les administrateurs peuvent gérer les utilisateurs de l'application et les projets,
- la partie traduction dans laquelle deux types d'utilisateurs (traducteurs et relecteurs) vont agir afin de traduire des fiches puis de les relire.

Ces fiches vont contenir les données provenant du MediaContent. Ces données pourront par exemple correspondre aux caractéristiques de différents outils de bricolage si l'entreprise cliente s'agit d'une entreprise commerciale. Dans l'interface de traduction, ces données seront stockées sous forme de deux tableaux côte à côte. Dans le tableau de gauche est stockée les données à traduire et dans le tableau de droite les champs à remplir dans la nouvelle langue.

Projet E-Translate : test2



Libellé Référence	Contenu Référence	Libellé traduit	Contenu traduit
Référence BIZ	BIZ 700 846	Viite	BIZ 700 846
Désignation longue (marketing)	Rallonge BIZSPOT	Pitkä kuvaus (markkinointi)	
Statut	6	Tila	
Série		Alavalikoima	
Svenskt E-nummer		Svenskt E-nummer	
Finskt SSTL		Finskt SSTL	
Norskt E-nummer		Norskt E-nummer	
Belgiskt BPP		Belgiskt BPP	
Début de vente		Myynnin aloituspäivä	
Fin de vente		Myynnin lopetuspäivä	

Fig. 2.1 : Capture d'écran de l'interface de traduction de l'application E-Translate

En ce qui concerne les projets, il s'agit simplement d'un regroupement de fiches. Ce sont les administrateurs qui s'occupent de créer un projet. Pour cela, ils vont indiquer un nom, une langue de référence (qui correspondra à la langue sur laquelle sera basée la traduction), une langue de traduction, un traducteur, un relecteur ainsi que la liste des fiches qui devront être traduites.

Une fois que le projet sera créé, ces fiches seront directement importées du MediaContent pour être transférer dans une base de donnée temporaire correspondant à celle utilisée par E-Translate. Le traducteur et le relecteur concernés par ce projet pourront ensuite s'occuper de la traduction. Une fois que leur travail sera fini, ces fiches pourront ensuite être exportées pour retourner dans la base de donnée du MediaContent.

Comme le montre la figure suivante, durant l'exécution d'un projet, une fiche va changer d'état. Lors de la création d'un projet, une fiche associée sera dans l'état « nouvelle ». Cela signifie que la fiche ne sera pas encore importée dans la base de donnée de l'E-Translate. Une fois que ce projet sera lancé, la fiche passera dans l'état « exportée » qui signifie que cette fois-ci la fiche sera dans cette base de donnée. Enfin, la fiche passera dans l'état « validée » lorsque l'administrateur aura accepté la traduction et la relecture d'une fiche.

On constate cependant dans ce schéma que l'état « exportée » peut être décomposé en plusieurs sous-états. Ces états vont représenter l'avancement du travail de traduction réalisé par les utilisateurs sur la fiche. Cette procédure d'entreprise est appelée un workflow.

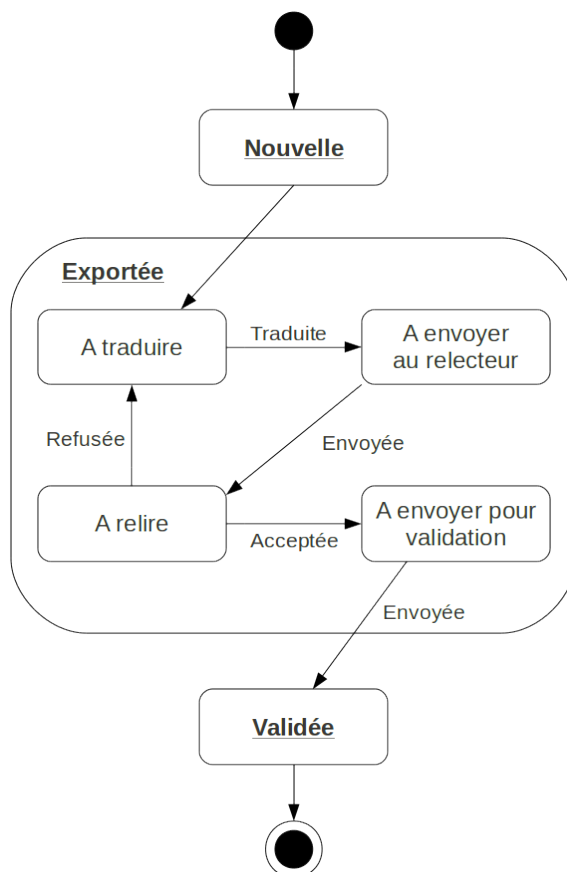


Fig. 2.2 : Les états d'une fiche au cours de sa traduction

Lorsque la fiche sera exportée, elle se retrouvera dans le sous-état « à traduire ». C'est pendant cette étape que le traducteur va pouvoir traduire les différents champs de la fiche. Une fois qu'il aura fini de la remplir, il pourra l'ajouter à la liste des fiches à envoyer correspondant à l'état « à envoyer au relecteur ». Lorsque le traducteur le souhaitera, il pourra envoyer la liste des fiches chez le relecteur. La fiche passera alors dans l'état « à relire ». Lors de l'envoi des fiches, un mail sera aussi envoyé au relecteur pour le prévenir. Une fois que le relecteur aura fini de relire la fiche il pourra :

- renvoyer la fiche au traducteur s'il considère que la traduction n'a pas été correcte. La fiche retournera alors à l'état « à traduire » ;
- accepter la fiche s'il considère que la traduction est convenable et par conséquent, la fiche sera ajoutée à la liste des fiches à envoyer (correspondant à l'état « à envoyer pour validation »)

Cette fois-ci, lorsque le relecteur enverra ses fiches relues, elles seront exportées de la base de données. Cela signifie que la fiche va quitter l'état « exportée ».

## 2.2.2. Structure de l'application

Comme expliqué précédemment, l'application peut être découpée en deux parties correspondant à la partie administration et la partie traduction. Nous allons à présent voir plus en détail les différentes fonctionnalités offertes aux utilisateurs ayant accès à ces parties.

### La partie administration

Cette partie de l'application apparaît dans le MediaContent sous forme d'onglet. Elle est accessible uniquement par les utilisateurs de type « administrateur ». Leur rôle est de s'occuper de toute la gestion de l'application. On peut identifier deux principaux types de gestion dans E-Translate : la gestion des utilisateurs et la gestion des projets.

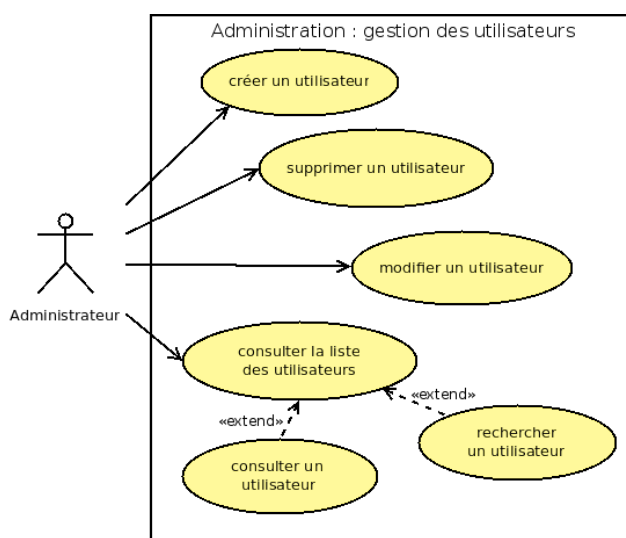


Fig. 2.3 : Cas d'utilisation – la gestion des utilisateurs

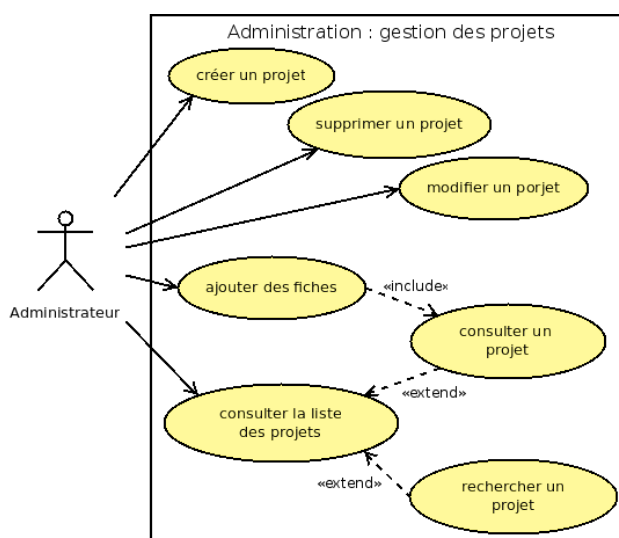


Fig. 2.4 : Cas d'utilisation – la gestion des projets

En ce qui concerne la gestion des utilisateurs, un administrateur a la possibilité de créer de nouveau utilisateur ou de les supprimer. Ces utilisateurs auront alors accès à la partie traduction de l'application. Ces utilisateurs seront destinés à traduire ou à relire les fiches qui leur seront affectées. Il est donc nécessaire de préciser à la création d'un utilisateur une langue dans laquelle l'utilisateur pourra travailler. L'administrateur a aussi la possibilité de consulter la liste des utilisateurs existant afin de pouvoir par la suite modifier les caractéristiques d'un utilisateur s'il le souhaite.

Pour ce qui est de la partie gestion des projets, l'administrateur va pouvoir créer de nouveaux projets auxquels sera lié deux langues (langue de référence et langue de traduction) et deux utilisateurs (traducteur et relecteur). Les fiches pourront être ajoutées par la suite à partir d'une liste contenant toutes les fiches contenues dans la base de données du MediaContent. Ces fiches ne pourront cependant être sélectionnées deux fois dans deux projets différents ayant la même langue de traduction. Une fois que ce projet sera créé, les fiches seront accessibles par les utilisateurs à partir de la partie traduction de l'application. Quand un projet sera terminé, l'administrateur pourra ensuite réexporter les fiches dans la base de données du MediaContent afin de clôturer le projet.

## La partie traductions

Contrairement à la partie administration, cette partie n'est pas directement intégrée dans le MediaContent car certains utilisateurs peuvent ne pas avoir de compte sur le MediaContent. Il s'agira alors de personnes externes à l'entreprise.

Dans cette partie, un projet va concerner deux utilisateurs. Un va jouer le rôle du traducteur et l'autre le rôle du relecteur. Cependant il faut bien comprendre qu'il n'y a pas deux types d'utilisateurs spécifiques. Cela signifie qu'un utilisateur pourra être traducteur dans un projet et relecteur dans un autre. Il est même possible d'être traducteur et relecteur du même projet. Les fonctionnalités entre un traducteur et un relecteur restent pratiquement les mêmes. La principale différence va agir au niveau du traitement d'une fiche.

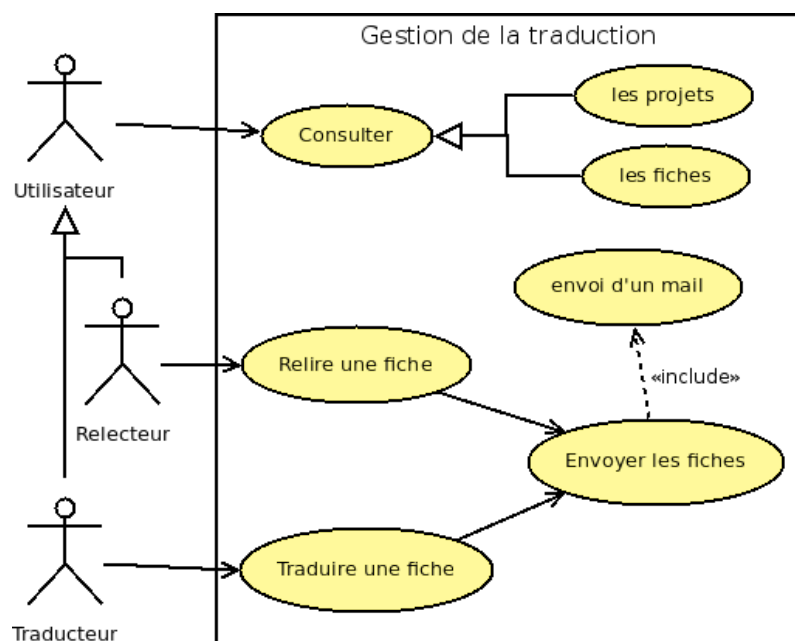


Fig. 2.5 : cas d'utilisation – gestion de la traduction

Un traducteur va donc pouvoir accéder à la liste de tous les projets dans lequel il est affecté. Il va donc pouvoir choisir le projet sur lequel il souhaite travailler pour ensuite afficher sa liste de fiches. Cette fois-ci, il va pouvoir choisir la fiche à traduire. L'interface de traduction va réunir deux tableaux côte à côte contenant chacun une colonne pour les libellés et une colonne pour les valeurs de ces libellés. Le tableau de gauche correspondra aux éléments d'une fiche dans la langue de référence et le tableau de droite contiendra les éléments de la même fiche dans la langue de traduction. Dans ce dernier tableau, les éléments seront vides mais éditables. Le traducteur aura donc la tâche de remplir tous les champs pour compléter une fiche. Lorsque toutes les fiches seront terminées, il pourra les adresser en bloc au relecteur.

Un relecteur va pouvoir lui aussi consulter les projets ainsi que les fiches qu'ils contiennent. Cependant, en ce qui concerne l'interface de travail, il ne pourra que relire le contenu des tableaux contenant les données des fiches. Dans cette interface, le relecteur pourra accepter ou refuser la traduction d'une fiche. Lorsqu'il l'accepte la fiche sera ajoutée dans la liste des fiches à envoyer à l'administrateur.



### 2.2.3. Description de la base de donnée

Nous allons maintenant nous intéresser à la base de données pour comprendre plus en détail comment sont stockées les données dans l'application E-Translate.

Cette application possède des utilisateurs, des projets, des fiches... Cela signifie donc que la base de donnée va comporter plusieurs tables pour stocker chacun de ces types d'informations. Cependant cette application doit être la plus générique possible car les données à stocker risquent d'être très différentes en fonction de l'entreprise pour laquelle l'application sera destinée. Il faut donc que la base de donnée utilise une méthode pour stocker les données qui n'imposera pas de devoir changer de structure pour chaque entreprise.

La méthode utilisée par Compilsoft pour pouvoir stocker tout type d'objet est la méthode « Entité – Attribut – Valeur ». Cette méthode est capable de transformer une base de données avec un grand nombre de table en une base de données composée uniquement de trois tables :

- une table « Entité » dans laquelle sont stockés les noms des tables ;
- une table « Attribut » qui possèdera les attributs des tables ;
- une table « Valeur » pour contenir toutes les valeurs des attributs.

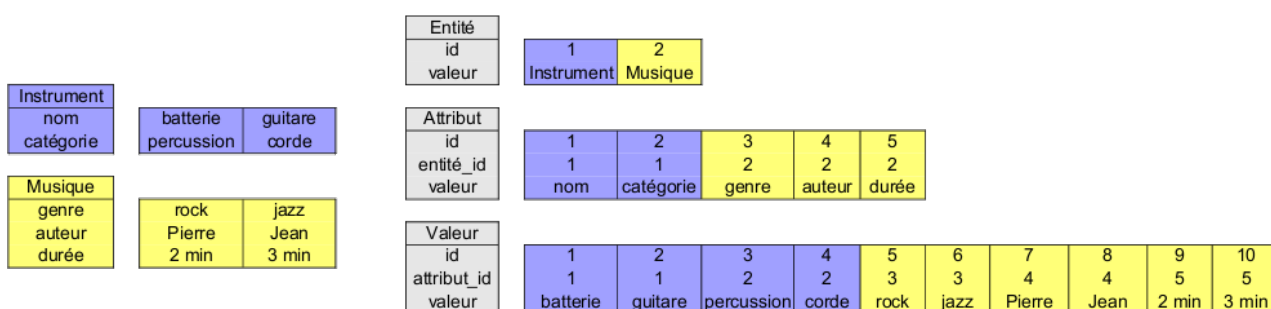


Fig. 2.6 : Exemple d'utilisation de la méthode « Entité – Attribut – Valeur »

Comme l'illustre l'image précédente, il est ainsi possible de stocker n'importe quelle structure de base de données en utilisant simplement ces 3 tables. La table « Entité » va contenir le nom des tables ainsi qu'un identifiant. La table « Attribut » va contenir les attributs de chacune des tables présente dans « Entité ». Il faudra cependant lier la valeur de l'attribut à la table grâce à « entité\_id ». Pour la table « Valeur » le principe est le même mais cette fois ci, ce sont les données des attributs qui sont stockées. Cette valeur sera liée à l'attribut grâce à « attribut\_id ».

Bien que cette méthode soit très pratique pour généraliser le stockage des données, il existe cependant quelques inconvénients. Ces tables ne possèdent que très peu de colonne mais vont au contraire avoir beaucoup de lignes. Pour la table « Entité » le nombre va être très faible. La table « Attribut » va déjà avoir un peu plus de lignes. Et pour finir, la table « Valeur » risque d'avoir un nombre très élevé de lignes. Il sera donc d'autant plus difficile voire même impossible de retrouver une information simplement en consultant la liste des données stockées dans ces tables.



En ce qui concerne la base de donnée de l'E-Translate, seules les fiches sont stockées en utilisant la méthode « Entité – Attribut – Valeur » car ce sont les seules informations qui risquent de changer d'une entreprise à l'autre. Les tables concernant les projets, les utilisateurs et les langues restent les mêmes.

Voici une représentation de la base de donnée de l'application E-Translate. Cette représentation n'intègre pas toutes les tables qui existent dans l'application (elle en comporte une trentaine) mais seulement celles qui sont utiles pour comprendre comment sont stockées les données les plus importantes.

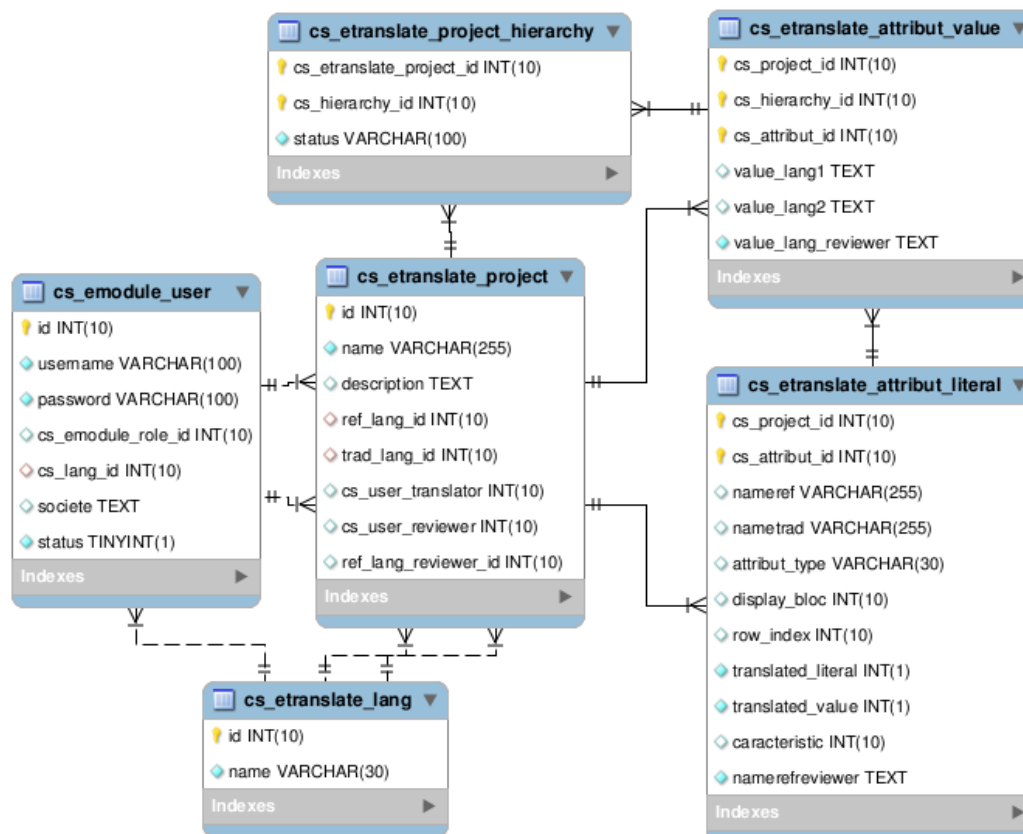


Fig. 2.7 : Diagramme représentant la base de donnée de l'ancien E-Translate

La table dans laquelle est stockée les utilisateurs s'appelle cs\_emodule\_user. Elle contient le login et le mot de passe de l'utilisateur mais également un statut pour déterminer si l'utilisateur est administrateur ou simple utilisateur. L'utilisateur possède aussi un identifiant vers la table cs\_etranslate\_lang qui correspond à la table contenant les langues. Il ne peut donc travailler que sur des projets dont la langue de traduction est la sienne.

Un projet est stocké dans la table cs\_etranslate\_project. On remarque que cette table est liée deux fois à cs\_emodule\_user et cs\_etranslate\_lang. Cette double liaison s'explique par le fait qu'un projet est lié à deux utilisateurs (un traducteur et un relecteur) et à deux langues (une langue de référence et une langue de traduction).



Les trois tables restantes sont utilisées pour stocker les fiches. La table `cs_etranslate_project_hierarchy` va jouer le rôle de la table « Entité », la table `cs_etranslate_attribut_literal` est la table « Attribut » et `cs_etranslate_attribut_value` correspond à la table « Valeur ».

On peut déjà remarquer que ces trois tables ne respectent pas le modèle standard « Entité – Attribut – Valeur » car la table des valeurs est directement liée à la table des entités. Cela s'explique par le fait que différentes fiches peuvent avoir des attributs en commun. Il ne faut donc pas qu'un attribut soit lié à une fiche si l'on ne veut pas avoir de redondance dans la table.

En ce qui concerne les tables des valeurs et des attributs, on constate que le champ contenant la valeur est présent deux fois. Ce double champ est nécessaire car il s'agit d'une application destinée à traduire toutes les informations des fiches. Il faut donc un champ pour le texte de référence et un champ pour le texte traduit.

Pour savoir à quel étape du workflow se trouve une fiche, il a été rajouté un attribut « state » dans la table des fiches. On peut ainsi savoir si une fiche se trouve par exemple dans l'état « en traduction » ou « validée ».

## 2.3. Description des outils de travail

Maintenant que l'explication de l'application existante a été faite, nous allons pouvoir nous intéresser aux outils mis à disposition par l'entreprise pour travailler. Ces outils concerneront les logiciels de programmation mais également des bibliothèques et des plateformes de travail.

### 2.3.1. Les logiciels

La programmation de la nouvelle application E-Translate s'est faite en PHP. Il est donc nécessaire d'utiliser un environnement de développement intégré (IDE) spécialisé pour ce langage afin de travailler convenablement. Pour cela, l'entreprise utilise une version de l'IDE Eclipse optimisé pour le langage PHP. L'utilisation de ce logiciel plutôt qu'un autre vient du fait que beaucoup d'applications développées par Compilsoft sont écrites en utilisant le langage Java. Eclipse étant destiné à la base pour ce langage, cela permet de pouvoir ouvrir des projets PHP et Java grâce au même logiciel.

Une des autres applications les plus utilisées est Tomcat. Cette application sert, pour les serveurs web, à faire tourner en arrière plan des sites Internet écrits en Java. Dans le cas de l'application E-Translate, celle-ci a été utilisée pour faire fonctionner l'application MediaContent afin de pouvoir faire communiquer l'E-Translate et le MediaContent sur le même ordinateur. L'avantage d'exécuter les applications localement est de pouvoir effectuer des tests sans déranger les autres utilisateurs de ces applications.

En ce qui concerne les projets, le logiciel de gestion de version nommé Subversion a été utilisé. Il permet à une équipe de développement de travailler sur un même projet sans avoir à se soucier de la mise à jour de ses fichiers. Tous les fichiers du projet sont synchronisés sur un serveur distant afin de permettre aux développeurs de toujours travailler avec les dernières versions des fichiers. Ce logiciel a été intégré directement dans Eclipse grâce à un plugin.

L'application E-Translate est intégrée dans le MediaContent sous forme d'un module. Il est donc nécessaire de bien hiérarchiser les fichiers afin de rendre l'application opérationnelle. De plus il faut compiler les fichiers java du MediaContent ainsi que générer un dossier compressé contenant tous les fichiers de





l'application. Pour cela, l'entreprise utilise un outil logiciel appelé Maven qui permet de générer automatiquement toutes ces étapes à partir d'un fichier de configuration écrit en XML. Il peut s'utiliser par la suite comme la commande Make sous Unix (c'est à dire en lançant simplement une requête de compilation).

### 2.3.2. Le framework PRADO

Concernant la programmation en PHP dans l'entreprise Compilsoft, une plateforme de travail (framework) nommée PRADO a été utilisée pour faciliter le développement des projets. Ce framework s'étend sur les langages PHP, HTML, CSS, Javascript et Ajax (c'est à dire les principaux langages utilisés sur le Web).

Le principal avantage de PRADO est de séparer la programmation des interfaces et la programmation des actions produites par les différents événements. Une interface correspond à la page HTML qui est générée dans le navigateur de l'utilisateur. L'écriture de cette page est donc découpée en deux fichiers :

- un fichier portant l'extension « .page » : Ce fichier est utilisé pour écrire l'interface de la page. Pour cela il utilise le langage HTML dans lequel peuvent être ajoutés quelques éléments spécifiques à PRADO.
- un fichier portant l'extension « .php » : Ce fichier correspond à un fichier PHP classique. Cependant, le fichier php doit contenir une classe du même nom que le fichier « .php » et « .page ». Mais pour fonctionner correctement, cette classe doit aussi hériter d'une classe appartenant au framework PRADO. Ce fichier peut alors être utilisé par exemple pour écrire le code des événements des différents boutons de l'interface décrite dans le fichier « .page ».

Ce framework offre aussi la possibilité de pouvoir généraliser certaine partie de l'interface grâce à des fichiers appelé « template ». Par exemple, un fichier template peut être utilisé pour définir l'en-tête et le pied d'une page HTML. Ainsi il n'est plus nécessaire de le ré-écrire dans chaque fichier de page.

## 2.4. La plateforme Activiti

L'équipe de développement a souhaité que la prochaine application E-Translate serve également de test pour l'intégration d'un nouvel outil de programmation au sein de la société Compilsoft. Cet outil nommé Activiti va permettre de gérer le workflow de l'application. Cela signifie que l'on n'aura plus besoin de programmer la gestion des workflows. On pourra ainsi se consacrer uniquement à la programmation des interfaces qui communiqueront avec Activiti.

Pour expliquer brièvement, Activiti est une plateforme de développement permettant de gérer des workflows en utilisant la notation standardisée de modélisation des procédures d'entreprise (BPMN). Cette notation permet de représenter des workflows sous forme de diagrammes faciles à comprendre pour un simple utilisateur. À partir de ces diagrammes, Activiti pourra par la suite exécuter le workflow grâce à une machine virtuelle qui tournera en arrière plan sur le serveur. L'application pourra ensuite communiquer avec cette machine virtuelle afin de progresser étape par étape dans le workflow.

Avant de s'attaquer à une explication un peu plus technique de la nouvelle application E-Translate, il est fortement recommandé de lire le document concernant Activiti en annexe 1.



### 3. Réalisation de l'application

Maintenant que l'explication de l'ancienne application E-Translate ainsi que les outils ont été présentés, nous pouvons entrer dans le vif du sujet : la nouvelle application E-Translate. Mais pour concevoir une application, il faut d'abord commencer par rédiger un cahier des charges qui reprendra la totalité des fonctionnalités qui seront développées dans l'application.

#### 3.1. Le cahier des charges

Ce document doit pouvoir décrire tout ce que l'application va pouvoir offrir à un utilisateur. C'est pourquoi il doit être claire et compréhensible par une personne non spécialisée. Cette partie du rapport ne va contenir qu'un résumé du cahier des charges afin d'éviter une lecture longue et fastidieuse. Il y sera néanmoins présenté les nouveautés qui seront apportées à la nouvelle application E-Translate, mais également la liste des différents éléments de cette application ainsi qu'une description du nouveau workflow.

##### 3.1.1. Les nouveautés apportées

Après avoir discuter de l'E-Translate avec notre chef de projet ainsi que d'autres développeurs, nous avons pu lister les nouvelles fonctionnalités que devra contenir la nouvelle application E-Translate.

Liste des modifications à apporter à la nouvelle application E-Translate	
L'ancien E-Translate	Le nouveau E-Translate
Lorsqu'un traducteur ou un relecteur décide d'envoyer ses fiches, un e-mail est également envoyé.	Pour limiter le nombre d'envoi d'e-mail, un système de notification permettra de savoir à partir de l'interface de l'application si de nouvelles fiches ont été reçues.
Les mots de passe sont stockés en clair dans la base de données.	Afin de mieux sécuriser les informations, les mots de passe des utilisateurs utilisent une méthode de cryptage efficace appelée « MD5 ».
Le workflow est géré par l'application. Le changement d'état d'une fiche est géré par une variable contenant le statut de la fiche dans la base de données.	Le workflow est géré par la plateforme Activiti. L'application E-Translate ne consistera qu'à interfacer le cycle de vie du workflow.
Un projet en cours est toujours modifiable.	Lorsqu'un projet est en cours, il ne peut plus être modifié.
Dans l'interface d'un traducteur ou d'un relecteur, les fiches d'un projet sont affichées dans une liste.	Pour améliorer la lisibilité des fiches d'un projet, elles sont stockées sous forme d'arborescences.
Lors de la création d'un utilisateur, une seule langue peut lui être attribuée.	Afin d'éviter la création de multiples comptes par un même utilisateur, il est maintenant possible d'ajouter plusieurs langues à un utilisateur.



Liste des modifications à apporter à la nouvelle application E-Translate	
L'ancien E-Translate	Le nouveau E-Translate
Certaines requêtes peuvent être longues. Il faut alors attendre que la page se recharge.	Lorsque les requêtes sont longues, l'application affichera une barre de progression afin que l'utilisateur puisse comprendre que l'application est en train de travailler.
Un projet permet de traduire tous les champs d'une fiche. C'est à dire les valeurs et les libellés.	Il existe plusieurs types de projet permettant de traduire soit les valeurs, soit les libellés soit une liste de données.
Lorsqu'un projet est en cours, il ne peut plus être arrêté.	Lorsqu'un projet est en cours ; l'administrateur a la possibilité d'interrompre l'exécution de celui-ci. Cependant une fois qu'il sera annulé, il ne pourra plus être relancé.
Si l'on souhaite créer un projet identique à un autre mais avec une langue différente, l'administrateur est obligé de tout réécrire.	Il est possible de dupliquer un projet pour gagner du temps lors de la création d'un nouveau projet semblable à un autre déjà existant.
Lorsqu'un relecteur envoie ses fiches à l'administrateur, celui-ci ne peut qu'exporter les fiches dans le MediaContent.	Lorsqu'un relecteur envoie ses fiches à l'administrateur, celui-ci peut les valider ou les refuser. S'il les refuse, il pourra les renvoyer au traducteur ou au relecteur.
Une fiche contient les champs des libellés et des valeurs associées dans les deux langues du projet.	Dans cette version, une fiche peut également contenir des images représentant l'objet décrit par la fiche.
Il existe deux types d'utilisateurs. Les utilisateurs classiques de l'application et les administrateurs.	Il existe un nouveau type d'utilisateur appelé « super-administrateur » qui aura la possibilité d'accéder à toute l'application et quelques autres fonctionnalités.



### 3.1.2. Les éléments de l'application

Pour bien comprendre comment fonctionne la nouvelle application E-Translate, il est important de connaître les éléments qui la composent. Cette partie va donc contenir une explication détaillée de ces différents éléments. Certaines caractéristiques de ces éléments auront déjà été citées lors de l'analyse de l'ancien E-Translate. Cette partie va permettre de rentrer plus en détails sur leur utilité.

#### Les fiches

Les fiches sont les éléments les plus importants de l'E-Translate car toute l'application repose sur elles. Chaque fiche correspond à un produit dont le libellé de ses caractéristiques et leurs valeurs peuvent être traduites. Cela signifie que les fiches seront les éléments sur lesquels vont travailler les utilisateurs de l'application. Les différentes caractéristiques qui composent une fiche sont :

Attribut	Description
nom	Le nom de la fiche permet à l'utilisateur de comprendre de quel produit il s'agit.
référence	La référence est unique par fiche. Elle permet donc d'identifier une fiche et ainsi de pouvoir différencier deux produits portant par exemple le même nom.
liste d'attributs	Les attributs d'une fiche correspondent aux divers caractéristiques que peut contenir un produit comme par exemple une taille, un diamètre, ...
liste de valeurs	Ces valeurs correspondent aux valeurs des attributs décrits précédemment.
image	Dans une fiche, l'image permet de visualiser le produit lié à la fiche.
état	L'état permet de connaître l'avancement du travail sur une fiche.

En ce qui concerne l'état de la fiche, son évolution va être liée au workflow. Voici la liste des états que va pouvoir prendre une fiche :

État	Description
en traduction	La fiche est en train d'être traduite par un traducteur
traduite	La fiche est prête à être envoyée au relecteur, le traducteur peut encore la relire
en relecture	La fiche est en train d'être relue par le relecteur il peut soit l'accepter soit la refuser. Dans ce dernier cas elle retournera au traducteur.
relue	La fiche est prête à être envoyée à l'administrateur pour une demande de validation.
en validation	La fiche est en train d'être validée par l'administrateur pour être intégrée au MediaContent. S'il subsiste des erreurs il peut la refuser.
validée	Les données de la fiches sont exportées dans le MediaContent



## Les projets

L'e translate est composé d'un ensemble de projet qui correspond à la mise en relation d'un ensemble de fiches à traduire avec des intervenants pour effectuer cette traduction. Il est possible d'associer un projet avec une date limite pour que les traductions soient effectuées sous une contrainte de temps. Tant que le projet n'est pas clos, il est possible d'ajouter des fiches au projet. Un projet se compose de :

Attribut	Description
nom	Le nom d'un projet est celui qui apparaît dans les interfaces affichant la liste des projets.
identifiant	Il permet de distinguer les projets même s'ils portent le même nom.
statut	Le statut est utilisé pour identifier dans quel état est le projet (nouveau, en cours, ...)
type	Le type de projet permet de déterminer quels champs des fiches seront traduits.
arbre	Lors de la création d'un projet, celui-ci est associé à un arbre qui sera utilisé par la suite pour ajouter des fiches au projet.
langue de référence	Cette langue correspond à la langue dans laquelle vont être récupérés les fiches du projet.
langue de traduction	Cette langue est, la langue dans laquelle seront traduites les fiches.
traducteur	Un utilisateur est associé au projet comme traducteur.
relecteur	Un deuxième utilisateur joue cette fois-ci le rôle du relecteur
date de création	Lorsqu'un projet est créé, la date est enregistrée.
date limite de traduction	Les fiches ne doivent pas rester en cours de traduction après cette date.
date limite de relecture	Les fiches ne doivent pas rester en cours de relecture après cette date.
ensemble de fiches	Cet ensemble de fiche contiendra les fiches sur lesquelles travailleront le traducteur et le relecteur.

Pour ce qui est du statut du projet, il peut être :

Statut	Description
nouveau	Un projet est « nouveau » lorsqu'il vient d'être créé.
en cours	Un projet est « en cours » lorsqu'il est démarré.
terminé	Un projet est « terminé » lorsque toutes ses fiches sont validées.
annulé	Un projet est « annulé » lorsque l'administrateur a interrompu son exécution.

En ce qui concerne le type de traduction d'un projet il existe :

Type de traduction	Description
libellés	Seul les libellés des fiches seront traduits.
valeurs	Seules les valeurs des fiches seront traduites.
listes	Ce type concerne la traduction de données stockées sous forme de liste.



### Les langues

Étant donné que l'application E-Translate est une application destinée à traduire des données, elle doit posséder des langues sur lesquelles les utilisateurs pourront travailler. Ces langues seront importées depuis le MediaContent par fichier XML. Une langue possède plusieurs caractéristiques :

Attribut	Description
libellé	Le libellé correspond au nom de la langue
code	Le code d'une langue est un acronyme du libellé. Par exemple « fr » pour la langue française.
drapeau	Le drapeau est une image représentant le drapeau du pays.

### Les utilisateurs

Les utilisateurs sont les acteurs de l'application. En fonction de leur statut, ils peuvent effectuer différentes actions dans l'application. Cependant, pour gérer efficacement le workflow, Activiti a besoin de connaître les utilisateurs. C'est pourquoi ils sont stockés dans la base de données d'Activiti. La base de données d'E-Translate ne fait que référencer les utilisateurs d'Activiti. Les informations associées au compte d'un utilisateur sont :

Attribut	Description
login	Le login correspond à l'identifiant de connexion de l'utilisateur.
mot de passe	Le mot de passe sera utilisé également à la connexion.
nom	Le nom de l'utilisateur.
prénom	Le prénom de l'utilisateur
mail	L'adresse e-mail de l'utilisateur qui sera utilisée pour la réception des messages.
statut	Le statut permet de définir les droits de l'utilisateur. (exemple : administrateur)
liste de langue	Cette liste de langue correspond aux langues que l'utilisateur sait pratiquer.

Le statut d'un utilisateur peut prendre trois valeurs différentes :

Statut	Description
super-administrateur	Un super-administrateur a accès à toute l'application.
administrateur	Un administrateur a accès à la partie administration de l'application.
utilisateur	Un utilisateur a accès à la partie traduction de l'application.

Le super-administrateur est un utilisateur particulier de l'application car il peut accéder à la totalité des fonctionnalités de l'application E-Translate. Il peut également utiliser d'autres fonctionnalités que les administrateurs et les utilisateurs classiques ne peuvent pas utiliser. Il peut par exemple afficher l'historique des actions effectuées sur l'application ou encore déverrouiller des fiches que des utilisateurs auraient malencontreusement bloquées. Ce type d'utilisateur n'est utilisé que par l'équipe de développement de Compilsoft afin d'effectuer des opérations de support pour les clients.



Un administrateur de l'application E-Translate travaille dans la partie administration de l'application. Il va s'occuper de créer de nouveaux utilisateurs (de type administrateur ou utilisateur) ou modifier le compte d'utilisateurs déjà existant. Il va aussi créer les projets de traduction ou les valider quand ceux-ci seront terminés.

L'utilisateur classique ne peut accéder qu'à la partie traduction de l'application. C'est ce type d'utilisateur qui peut être associé à un projet. Il ne peut donc travailler ou même visualiser que les fiches du projet le concernant. Cependant, en fonction du rôle qu'il joue dans un projet, le travail qu'il pourra effectuer sera différent. Si l'utilisateur est traducteur, il peut modifier les informations des fiches dans la zone de traduction. Si l'utilisateur est relecteur, il peut relire les fiches traduites pour ensuite les accepter ou les refuser.

## 3.1.3. Le workflow

Le workflow de la nouvelle application E-Translate est légèrement différent de celui de l'ancienne application. Cette fois ci, le diagramme le représentant utilise la notation standardisé de modélisation de procédure d'entreprise (BPMN). Pour bien comprendre le fonctionnement d'un diagramme BPMN, il est nécessaire d'avoir lu au préalable la partie 2 (Fonctionnement d'Activiti) de l'annexe 1 (Activiti).

Dans cette nouvelle version, le workflow intègre quelques nouveautés :

- Il est maintenant possible de créer un projet sans relecteur.
- Un administrateur peut refuser une fiche et choisir de la renvoyer au traducteur ou au relecteur.
- La relecture et la traduction possèdent une date limite d'exécution.

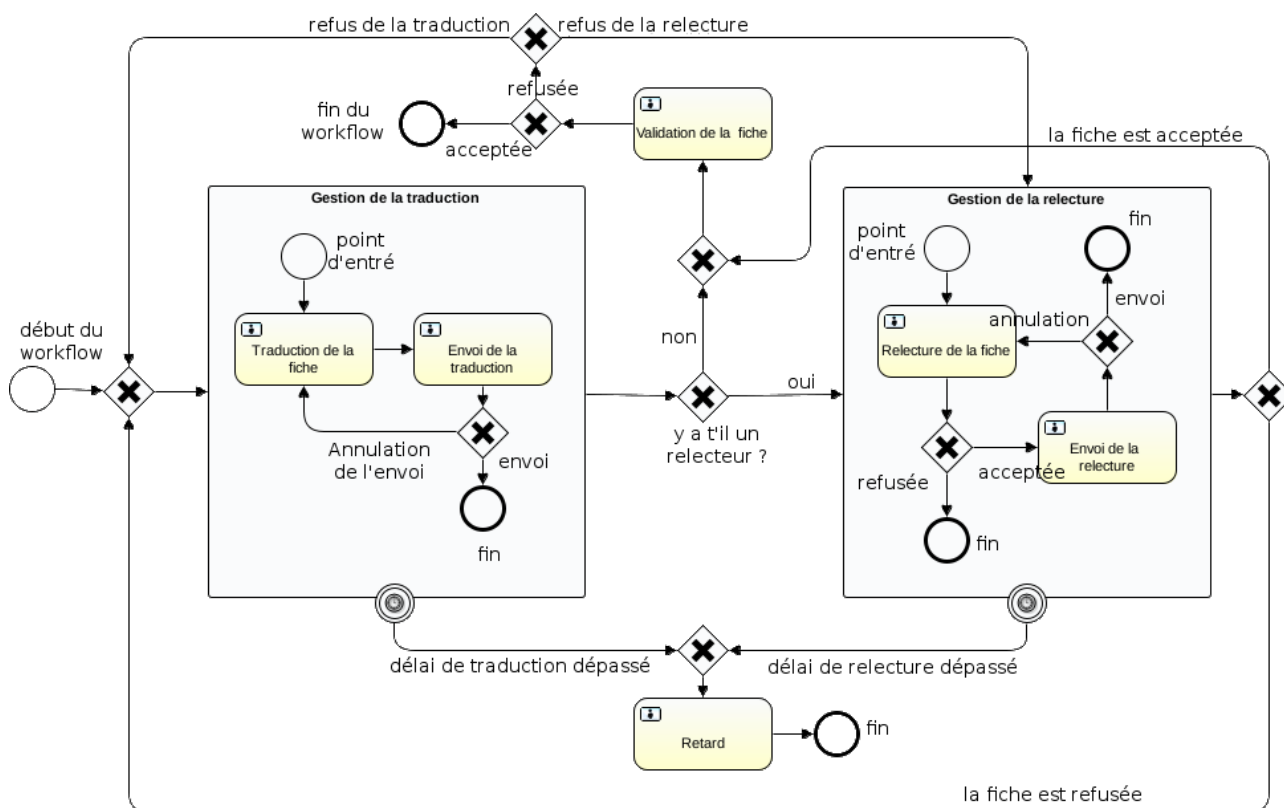


Fig. 3.1 : Diagramme BPMN représentant le workflow du nouveau E-Translate



On remarque que le workflow est composé de deux sous-processus appelés « Gestion de la traduction » et « Gestion de la relecture ». Les tâches qui sont à l'intérieur de ces sous-processus seront respectivement assignées au traducteur et au relecteur. Les autres tâches du processus globale seront affectées à l'administrateur.

Lorsque le processus va démarrer, il va commencer par appeler le sous-processus « Gestion de la traduction ».

### La gestion de la traduction

Lorsque ce sous-processus va commencer, la première tâche à être exécutée sera la « Traduction de la fiche ». Lorsque le workflow sera à cette étape, le traducteur va pouvoir travailler sur la fiche afin de traduire ses attributs. Cette tâche ne se terminera qu'une fois que le traducteur aura cliqué sur le bouton « À envoyer ».

Une fois que cette tâche sera terminée, le workflow passera à la tâche suivante nommée « Envoi de la traduction ». Lorsqu'une fiche est dans cet état, elle se trouve dans la liste des fiches prêtes à être envoyées. Le traducteur aura alors deux choix possible : soit il annule l'envoi de la fiche et le processus qui lui est liée retournera dans l'état « Traduction de la fiche » ; soit il décide d'envoyer la fiche et le sous-processus « Gestion de la traduction » se terminera.

Une fois que le sous-processus « Gestion de la traduction » est terminé, le workflow va pouvoir passer à la suite. Une fois que le workflow aura quitté le sous-processus, il pourra aller directement à la tâche « Validation de la fiche » si aucun relecteur n'a été affecté au projet. Cependant, si un le projet possède un relecteur, le workflow entrera dans le sous-processus « Gestion de la relecture ».

### La gestion de la relecture

Ce sous-processus commencera par la tâche « Relecture de la fiche ». C'est pendant cette tâche que le relecteur devra relire la fiche pour ensuite décider si elle est acceptée ou refusée. S'il refuse la fiche, le workflow quittera alors le sous-processus de gestion de la relecture et retournera directement au sous-processus de gestion la traduction. S'il accepte la fiche, le workflow passera alors à la tâche « Envoi de la relecture ».

Cette tâche fonctionne de la même façon que la tâche « Envoi de la traduction ». Cela signifie que le relecteur pourra annuler l'envoi pour que le workflow retourne à la tâche « Relecture de la fiche ». Mais cette fois ci, lorsqu'il enverra la fiche, le workflow ira obligatoirement à la tâche « Validation de la fiche ».

Durant cette tâche l'administrateur va pouvoir s'occuper de la validation de la fiche. Pour cette tâche trois choix sont possible :

- soit, l'administrateur décide de renvoyer la fiche au traducteur : le workflow retournera alors au sous-processus de gestion de la traduction ;
- soit, il décide de renvoyer la fiche cette fois ci au relecteur : le workflow retournera alors au sous-processus de gestion de la relecture ;
- soit, il décide d'accepter la fiche et par la même occasion il terminera le workflow.

En ce qui concerne la tâche « Retard », elle sera automatiquement démarrée lorsque la limite de traduction ou de relecture sera dépassée. L'administrateur pourra alors envoyer un mail aux personnes concernées.



## 3.2. Analyse de l'application

Maintenant que l'on sait ce que doit faire l'application, nous allons pouvoir étudier de façon un peu plus technique la structure qu'il va falloir mettre en place pour l'application ainsi que la base de données dans laquelle seront stockées toutes les informations relatives à l'application. Mais tout d'abord, intéressons nous à la place qu'aura l'application au sein du MediaContent.

### 3.2.1. Déploiement de l'application auprès du MediaContent

L'application E-Translate est avant tout destinée à être utilisée comme un module de traduction au sein d'une application MediaContent qui s'occupe de gérer du contenu. Cela signifie donc que l'E-Translate va devoir communiquer avec le MediaContent afin de récupérer les données qui seront par la suite utilisées pour la traduction. Mais l'E-Translate va aussi devoir communiquer avec une autre plateforme nommée Activiti. Celle-ci s'occupera de l'exécution des workflows de l'application.

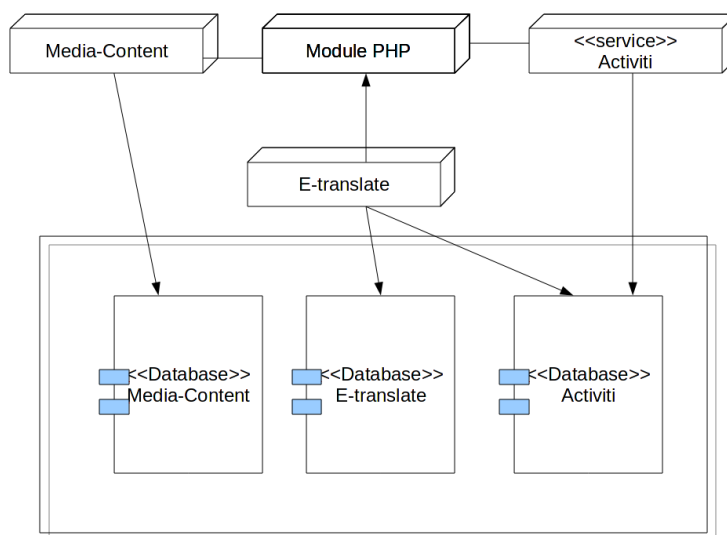


Fig. 3.2 : Diagramme de déploiement de l'E-Translate

Comme le montre ce diagramme, l'application E-Translate appartient à l'ensemble des modules PHP qui sont associés à l'application MediaContent. Cependant l'E-Translate ne peut pas récupérer des données directement dans la base de donnée du MediaContent. La méthode utilisée pour ces données est d'utiliser des fichiers XML. L'avantage de cette méthode est que si le MediaContent décide de changer de base de donnée, le module E-Translate fonctionnera toujours.

En ce qui concerne Activiti, ce module est actuellement utilisé uniquement par le module E-Translate mais il sera par la suite utilisé par tous les modules fonctionnant avec un workflow. C'est pourquoi sur le diagramme, ce service n'est pas lié directement à « E-Translate » mais à « Module PHP ». Cependant cette fois ci, le module E-Translate communique directement avec la base de données d'Activiti. Ceci s'explique par le fait que le web service d'Activiti est incomplet. Il n'est pas possible, par exemple, de créer un nouvel utilisateur. Mais ce problème sera réglé par la suite car un module faisant sur-couche à Activiti sera créé afin d'éviter à avoir à modifier la base de données d'Activiti.

### 3.2.2. La base de donnée

La base de donnée de l'E-Translate par rapport à celle de l'ancien E-Translate a quelque peu changée. Lors de l'analyse de l'ancienne application, la base de donnée a déjà été présentée. Cependant cette présentation ne contenait pas toutes les tables car elles étaient trop nombreuses et pas toutes utilisées par l'application. C'est pourquoi la nouvelle application possède une nouvelle base de donnée composée uniquement des tables utiles et qui intègre également quelques nouveautés.

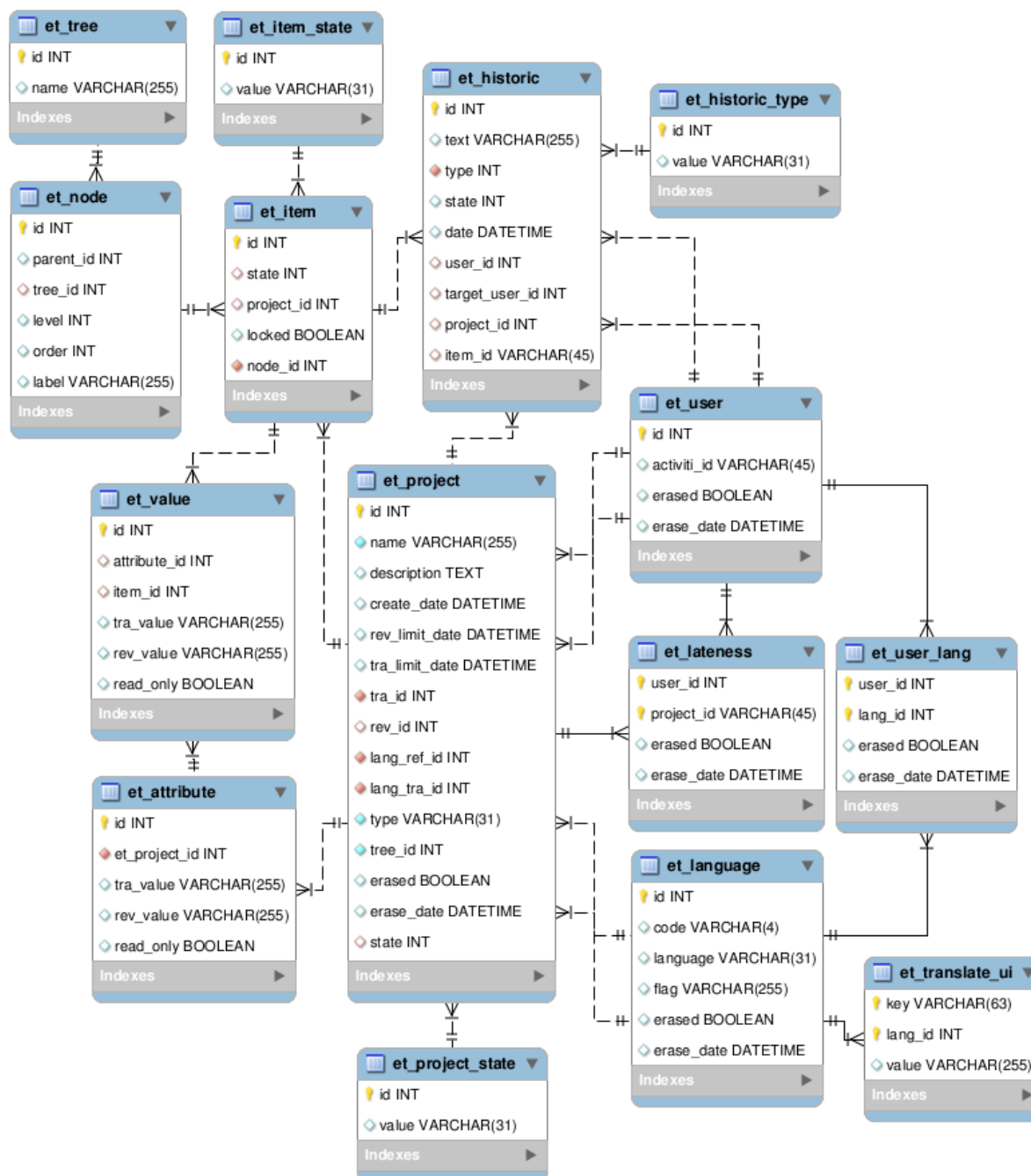


Fig. 3.3 : (Merise) Base de donnée de la nouvelle application E-Translate



Dans ce diagramme, on peut remarquer que les tables qui concernent les projets les fiches et les utilisateurs sont liées de la même façon que dans l'application E-Translate précédente. La table `et_project` contient les projets et la table `et_user` les utilisateurs. Les fiches sont toujours stockées en utilisant la méthode « Entité – Attribut - Valeur » dans les tables `et_item`, `et_attribute` et `et_value`. Bien que ces tables ressemblent à celle de l'ancien E-Translate, certains attributs ont changé.

### Les fiches

La table des fiches (`et_item`) possède maintenant un statut (`state`) pour savoir dans quel état se trouve la fiche. Par exemple, une fiche pourra être dans l'état « traduction en cours » ou « envoi de la relecture ». La liste de ces statuts est stockée dans la table `et_item_state`.

Une fiche possède également un booléen (`locked`) indiquant si la fiche est verrouillée ou non. Le verrou d'une fiche sera utilisé pour éviter que deux utilisateurs (connectés avec le même compte) ne puissent modifier la fiche en même temps.

Dans la base de données la fiche connaît aussi l'identifiant d'un nœud de l'arborescence (`node_id`). C'est cette arborescence qui a été utilisée pour fabriquer les fiches lors de la création d'un projet. A chaque nœud de l'arbre est associée une fiche.

Un autre attribut a aussi été ajouté cette fois-ci dans les tables des valeurs (`et_value`) et des attributs (`et_attribute`). Cet attribut (`read_only`) est de type booléen et permet de pouvoir savoir si un champ est modifiable ou pas pendant la traduction. Par exemple, une valeur numérique n'a pas besoin d'être traduite dans d'autres langues. Pour cette valeur, le booléen indiquera que le champ est en lecture seule.

### Les arbres

Les arbres sont stockés dans deux tables : la table `et_tree` qui contient le nom de l'arbre et la table `et_node` qui contient les nœuds. L'arborescence va se former simplement grâce au fait qu'un nœud connaisse son père (`parent_id`). La table `et_node` possède aussi d'autres arguments permettant de connaître son rang dans l'arbre (`level`), sa position parmi les nœuds fils d'un nœud père (`order`) ainsi qu'un libellé (`label`).

### Les projets

La table qui contient les projets (`et_project`) est assez similaire à celle de l'ancien E-Translate. On y retrouve de nombreux liens avec d'autres tables comme les utilisateurs, les langues, les fiches. Cependant cette table va apporter deux nouveautés : l'ajout de deux dates correspondant aux dates limites de traduction et de relecture ainsi que l'ajout d'un statut. Ce statut qui est contenu dans la table `et_project_state` permet de savoir dans quel état se trouve un projet (par exemple : nouveau, en cours, annulé ou terminé).

### Les utilisateurs

Normalement les utilisateurs ne sont pas stockés dans cette application mais dans Activiti. Cependant, pour simplifier certaine requête, une table (`et_user`) va contenir le login de l'utilisateur dans Activiti. Un utilisateur va aussi pouvoir être lié à plusieurs langues par l'intermédiaire de la table `et_user_lang` qui permet d'associer une langue à un utilisateur.



### Les langues

Les langues de l'application sont stockées dans la table `et_language`. Ces langues sont importées du MediaContent à partir d'un fichier XML. Le champ `flag` correspondant au drapeau contiendra simplement un lien vers l'image du drapeau.

### Le texte de l'application

L'application E-Translate peut être utilisée par des utilisateurs ne connaissant pas le français. C'est pourquoi la totalité du texte écrit dans l'application (comme par exemple le nom des menus) est stockée dans la table `et_translate_ui` de la base de données. Cette table contiendra un mot clé (`key`) pour l'identifier dans l'application ainsi que l'identifiant d'une langue (`lang_id`). Le dernier champ (`value`) va contenir le texte qui sera affiché dans l'application dans la langue définie par `lang_id`.

### Les retards

Lorsqu'un traducteur ou un relecteur ne termine pas son travail avant la date limite, une ligne sera rajoutée dans la table gérant les retards (`et_lateness`). Un retard va donc lier un utilisateur et un projet.

### L'historique

Pour conserver une trace des actions qui ont été faites par les utilisateurs dans l'application, il existe une table nommée `et_historic`. Les actions de cette table seront liées à l'utilisateur qui a effectué l'action, l'utilisateur qui a subi l'action, le projet concerné et la fiche. Il est cependant possible qu'un ou plusieurs de ces éléments ne soient pas liés à une action. Cette table possède aussi un type d'action stocké dans la table `et_historic_type` (par exemple e-mail, message, connexion, ...) ainsi qu'un statut pour savoir si l'action a rencontré une erreur.

Lors de l'utilisation de l'application, il peut parfois arriver qu'un utilisateur supprime une donnée puis se rend compte qu'il a commis une erreur et souhaiterait récupérer la donnée perdue. C'est pourquoi les données de certaines tables ne sont jamais supprimées pour permettre ainsi leur restauration. Lorsqu'une donnée sera supprimée, un booléen (`erased`) permettra d'indiquer que la donnée a été supprimée. Elle ne sera alors plus affichée dans l'application mais sera toujours présente dans la base de données. La date de suppression (`erase_date`) sera également conservée pour garder une trace de l'action.



### 3.2.3. La structure de l'application

Comme il a été expliqué lors de la présentation des outils de travail, l'application E-Translate est développée en utilisant le langage PHP ainsi que le framework PRADO. L'avantage de ce langage est d'être orienté objet. Il est donc possible de réaliser un diagramme de classe de l'application.

Voici le diagramme représentant l'application. Il n'intègre pas toutes les méthodes ni toutes les classes mais uniquement celles qui sont utiles pour comprendre la structure de l'application.

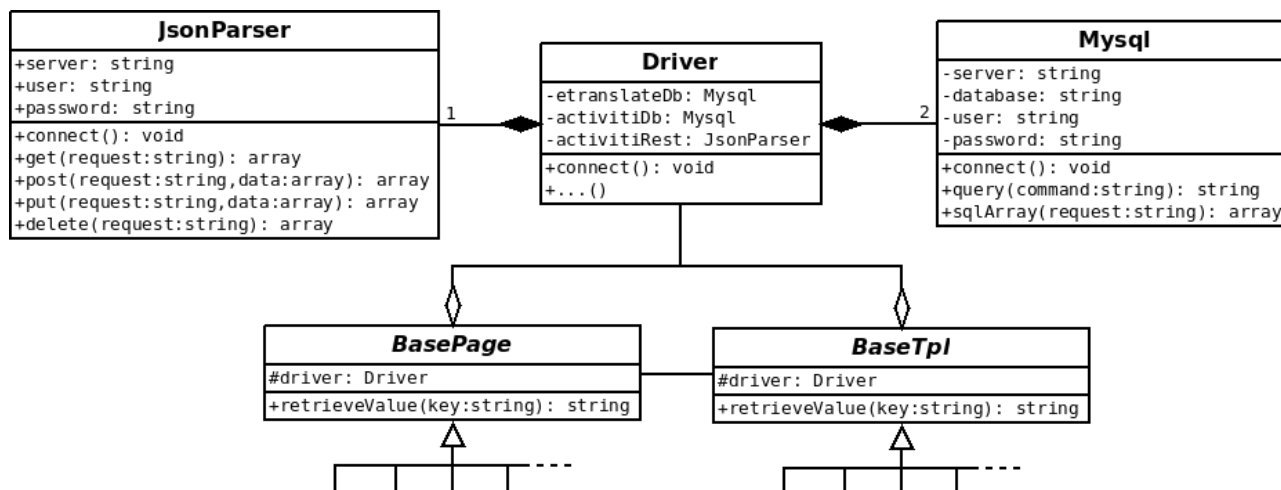


Fig. 3.4 : Diagrammes de classe simplifié de l'application E-Translate

Avant toute chose il faut rappeler que lorsque l'on utilise le framework PRADO, pour chacune des pages de l'application est associée une classe PHP. Ces pages contiendront divers informations en fonction des composants que la page va afficher mais il existe cependant des informations identiques pour chacune des pages de l'application. C'est pourquoi il est nécessaire de créer une classe mère dont chacune des pages héritera. La classe mère nommée **BasePage** va donc pouvoir contenir toutes les informations communes à chacune des pages. Il en est de même pour les patrons des interfaces de l'application qui hériteront de la classe **BaseTpl**.

L'application doit pouvoir communiquer avec la base de donnée. Pour cela, il suffit de créer une classe nommée **Mysql** qui va ainsi pouvoir permettre d'établir la connexion avec la base de donnée. Cependant cette connexion nécessite de préciser le nom d'un serveur, le nom de la base de donnée, le nom de l'utilisateur et son mot de passe. Il ne restera plus qu'à écrire des requêtes dans le langage SQL et de les envoyer à la base par l'intermédiaire de la méthode `query()` de cette classe. Si les requêtes envoyées à la base retournent des informations, elles pourront être récupérées par la méthode `sqlArray()`.

En ce qui concerne la communication avec Activiti, elle se fera par l'intermédiaire du web service nommé **Activiti REST**. L'utilité de ce web service est de pouvoir effectuer des actions sur les processus d'Activiti en utilisant un langage web autre que le langage Java (qui est le langage utilisé par Activiti). Ce web service va s'utiliser de la même manière que le protocole HTTP. C'est à dire qu'il suffira d'envoyer des requêtes de type GET, POST, PUT ou DELETE pour effectuer des actions. Ces requêtes permettent par exemple de demander à Activiti combien de processus sont en fonctionnement, ou encore récupérer les informations d'une tâche à partir de son identifiant. Le web service répondra à toutes ces requêtes en utilisant le format JSON pour structurer les données transmises. La classe de l'application qui s'occupe de la communication avec Activiti REST s'appelle **JsonParser**. Comme la classe **Mysql**, elle nécessite d'établir une connexion en



utilisant un nom de serveur, un nom d'utilisateur et un mot de passe. La communication se fera ensuite par les méthodes `get()`, `post()`, `put()` et `delete()` correspondant à chacun des types de requête HTTP.

L'application a donc la possibilité de communiquer avec la base de données et avec Activiti mais l'écriture des requêtes est assez compliquée et souvent utile dans plusieurs pages. C'est pourquoi il existe une autre classe nommée `Driver` qui fait office de proxy entre les classes de communications et les classes des interfaces. Cela signifie que toutes les requêtes qui peuvent être utiles pour les pages de l'application seront écrites dans cette classe. Le `Driver` possède donc une instance de la classe `JsonParser` pour communiquer avec Activiti et une instance de la classe `Mysql` pour communiquer avec la base de données. On remarque cependant que le `Driver` possède aussi une autre instance de la classe `Mysql` permettant de communiquer avec la base de données d'Activiti. Cette dernière est nécessaire car le web service d'Activiti ne permet pas de tout faire.

### 3.3. La méthode Scrum

L'explication du fonctionnement de l'application est terminée. Nous allons maintenant pouvoir passer au détail de la programmation. Mais avant cela, pour bien comprendre comment a été structuré le temps de travail lors de la programmation, il est nécessaire d'avoir quelques connaissances sur la méthode Scrum.

Scrum est une méthode de gestion de projets appartenant à la famille des méthodes agiles. Ces méthodes impliquent au maximum le demandeur (le client) et permettent une grande réactivité à ses demandes. Elles visent la satisfaction réelle du besoin du client en priorité aux termes d'un contrat de développement. (extrait de l'article « Méthode agile » de Wikipédia)

Pour optimiser la communication avec les clients, la méthode Scrum propose de découper la création d'une application en plusieurs itérations (appelées Sprint). Cela signifie qu'un client n'aura pas à attendre pendant une longue période le temps que l'équipe de développement s'occupe de créer entièrement l'application. En utilisant la méthode Scrum, le client ne patientera que le temps de réalisation d'un sprint avant d'avoir quelque chose de concret. Cela va ainsi permettre d'éviter de rompre trop longtemps la communication entre le client et le pôle de développement.

#### 3.3.1. L'organisation d'un projet

Comme le montre la figure 3.5, un projet sera composé de plusieurs releases. Une release correspondra à une version opérationnelle d'un projet. Un client peut décider d'arrêter le développement de l'application après qu'une release soit terminée si il considère que l'application créée est suffisante.

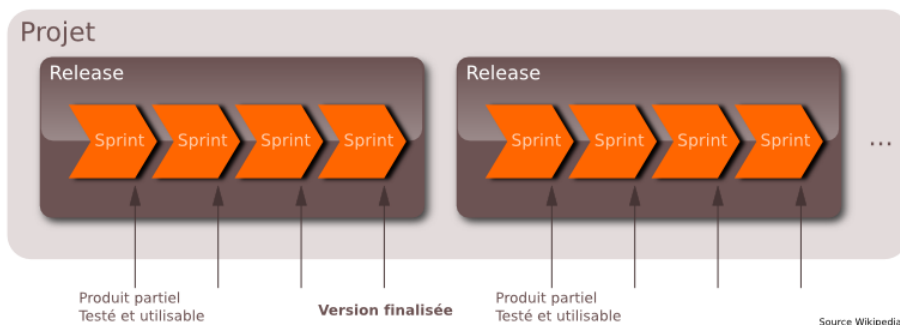


Fig. 3.5: Composition d'un projet utilisant Scrum (source <http://www.makina-corpus.com/>)

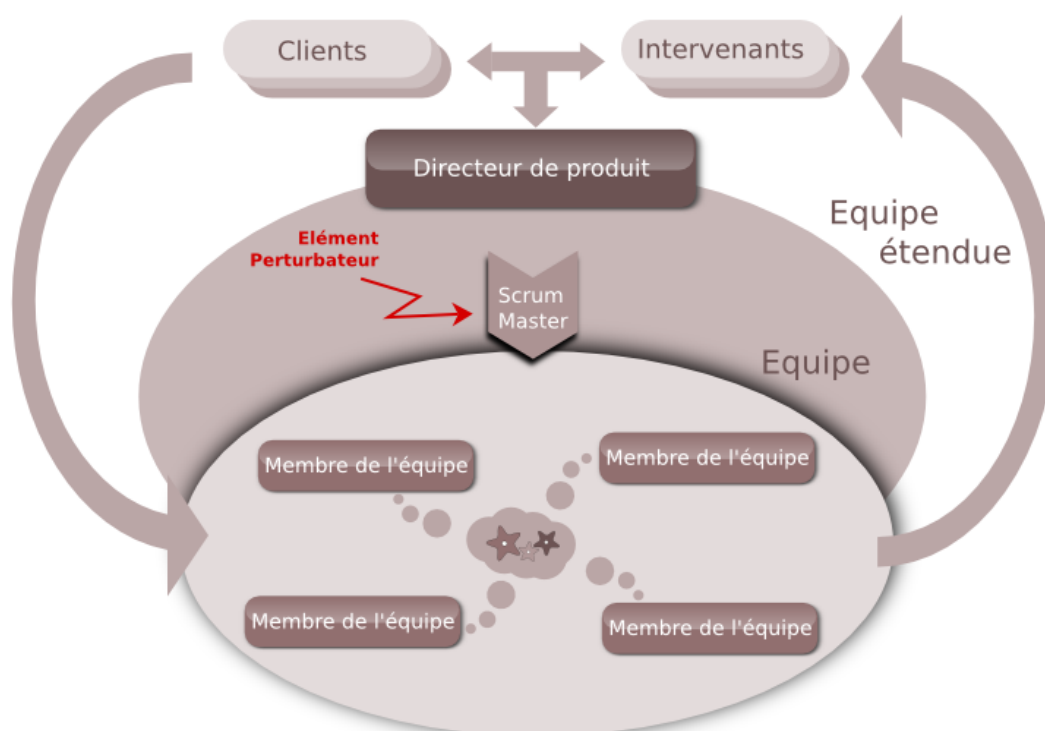


Une release est quant à elle composée de plusieurs sprints. Un sprint aura une durée fixe qui peut être comprise entre 2 et 4 semaines. C'est pendant la durée d'un sprint que l'équipe de développement s'occupe de créer une partie de l'application. Lorsqu'un sprint sera terminé, l'application contiendra les nouvelles fonctionnalités ajoutées. Cela signifie qu'à la fin de chaque sprint, l'application s'agrandira tout en restant toujours fonctionnelle.

### 3.3.2. L'organisation de l'équipe

Pour la création d'un projet, Scrum définit 3 rôles :

- **Le directeur du produit** : La personne qui porte ce titre s'occupe de la communication avec les clients ainsi que les intervenants de l'application. C'est cette personne qui décide les fonctionnalités que doit posséder l'application.
- **Le Scrum Master** : Cette personne est le responsable de la méthode Scrum. Son rôle consiste à guider les développeurs du projet et s'occuper de la communication avec le directeur du produit.
- **L'équipe de développement** : Elle est constituée de l'ensemble des développeurs travaillant sur le projet. Cette équipe va travailler sur l'implémentation des demandes qui ont été faites pour le sprint actuel.



Source Wikipedia

Fig. 3.6: Organisation de l'équipe (source <http://www.makina-corpus.com>)



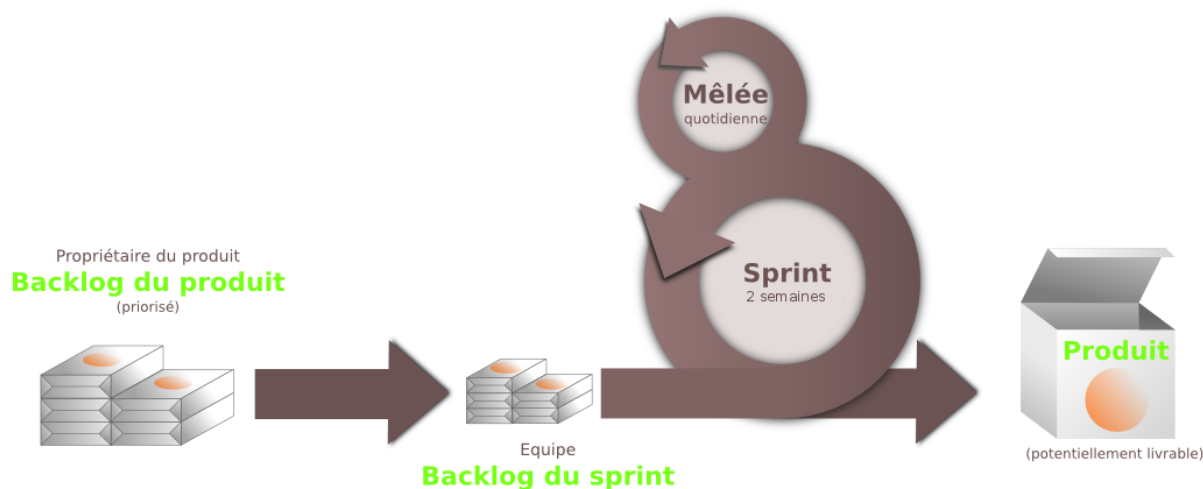
### 3.3.3. Le fonctionnement de Scrum

Lors de la création d'un nouveau projet, le directeur du produit va communiquer avec les clients afin d'établir la liste complète des fonctionnalités que l'application devra posséder. Ces fonctionnalités seront écrites dans un document appelé Backlog du produit. Ce document pourra être modifié à n'importe quel moment par le client pour intégrer de nouvelles fonctionnalités ou en retirer.

Une fois que le projet sera créé, un premier sprint va pouvoir commencer mais avant de le démarrer, il va falloir que le Scrum Master et l'équipe de développement mettent en place un autre document appelé Backlog du sprint. Ce document va contenir quelques fonctionnalités du Backlog du produit qui seront ensuite développées lorsque le sprint commencera. Contrairement au Backlog du produit, le Backlog du sprint ne pourra pas être modifié en cours de route. Les nouvelles fonctionnalités à intégrer seront ajoutées lors de la création du sprint suivant.

Un sprint possède également une mêlée quotidienne qui permet de faire un point entre les développeurs. C'est donc une courte réunion qui dure environ 15 minutes pendant laquelle chacun des développeurs explique ce qu'il a fait hier et ce qu'il va faire aujourd'hui.

À la fin d'un sprint, le directeur du produit pourra montrer le produit qui vient d'être créé au client afin qu'il puisse vérifier si l'application correspond à ses attentes. Il pourra ensuite décider de continuer de faire évoluer l'application en débutant un nouveau sprint ou de s'arrêter ici.



Source Wikipedia  
(Copyright - Mountain Goat Software)

Fig. 3.7 : Illustration du fonctionnement de la méthode Scrum (source <http://www.makina-corpus.com>)





### 3.4. Détail de la programmation

Le détail de la programmation de l'application ne concernera que le travail que j'ai effectué. Le travail de l'autre membre de l'équipe (Jérémy DENIS) ne sera donc pas expliqué.

La programmation de l'application E-Translate a été faite en utilisant la méthode Scrum. Cela signifie que la création de l'application a été découpée en plusieurs sprints. Cependant, lors du commencement de la programmation de l'application, il ne restait plus que 3 semaines de stage. C'est pourquoi il a été décidé de diviser la programmation en 3 sprints d'une semaine.

#### 3.4.1. Le premier sprint

Le premier sprint de l'application est le plus compliqué à mettre en place car l'équipe part de zéro et doit distribuer une application fonctionnelle à la fin de la semaine. Il est donc important lors de la création du backlog du sprint de ne pas être trop gourmand pour être sûre d'avoir tout fini avant la date limite.

Durant ce premier sprint, la majorité du travail effectué tournait autour de la gestion du workflow. Voici la liste des choses qui ont été réalisées durant ce premier sprint :

Travail à réaliser	Développeur
Création de la base de donnée	Cyrille
Création du workflow	Cyrille
Création des drivers	Jérémy
Programmation des interfaces d'affichage d'un projet	Cyrille
Programmation des interfaces de gestion d'une fiche	Jérémy

#### Création de la base de donnée

La base de donnée a été mise assez rapidement en place étant donnée que l'analyse de sa structure avait été faite auparavant. Il ne restait plus qu'à y insérer des données pour pouvoir effectuer des tests avec l'application. Pour l'instant, les données sont insérées manuellement dans la base de données mais lorsque le système d'importation des données par fichier XML sera mis en place, les données seront ajoutées automatiquement.

#### Création du workflow

De la même façon que la base de donnée, le workflow a été écrit lors de l'analyse de l'application. Toutes les tâches sont donc déjà créées. Cependant, seul la forme du workflow a été mise en place. Il est encore nécessaire de créer et de configurer les variables qui sont associées à chacune des tâches. Une description technique du workflow est disponible en annexe 3.



### Programmation des interfaces d'affichage d'un projet

Cette partie de la programmation de l'application concerne les interfaces de visualisation d'un projet. Mais en fonction du type d'utilisateur, la visualisation d'un projet ne sera pas la même. C'est pourquoi il existe 3 interfaces de visualisation :

- la visualisation d'un projet par un administrateur (classe : Project),
- la visualisation des fiches d'un projet en cours de traduction (classe : WorkProject),
- la visualisation des fiches d'un projet prête à être envoyée (classe : ViewProject).

Ces interfaces ont beaucoup de caractéristiques communes. Il peut donc être judicieux de les faire hériter d'une classe mère qui contiendra ces caractéristiques. Cependant, bien qu'il soit possible de faire hériter la classe PHP d'une page PRADO, il n'est pas possible de faire hériter le code de l'interface en elle-même. Une autre solution consisterait à utiliser le système de patron de page de PRADO afin d'éviter au maximum d'avoir deux fois le même code pour les interfaces des projets.

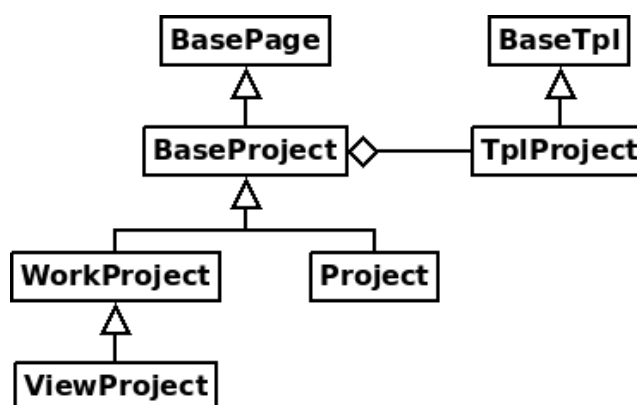


Fig. 3.8 : Diagrammes de classe des pages concernant les projets

Comme le montre la figure 3.8, les trois pages concernant la visualisation des projets héritent de la classe **BaseProject**. Les informations partagées par ces trois classes sont les informations qui sont stockées dans la base de données. Ces informations concernent le projet ainsi que les langues, les utilisateurs et les fiches qui sont associées à ce projet.

La classe **ViewProject** qui concerne la visualisation des fiches prêtes à être envoyées hérite directement de la classe **WorkProject** car ces deux pages sont très similaires. Dans ces deux interfaces l'affichage des informations est la même. Le seul changement qu'apporte **ViewProject** par rapport à **WorkProject** est une colonne de plus dans le tableau listant les fiches du projet. Cette colonne permet de pouvoir sélectionner les fiches à envoyer.

La différence entre **WorkProject** et **Project** concerne cette fois-ci l'affichage des fiches du projet et également les informations du projet. La page **Project** ajoute une colonne permettant de voir le statut des fiches ainsi qu'un tableau contenant l'avancement du projet (voir figure 3.9).

Tous les textes affichés dans l'application, comme par exemple les titres des tableaux ou les libellés des informations du projet, sont stockés dans la base de données afin d'être interchangeables en fonction de la langue de l'application choisie par l'utilisateur.



Identifiant :	1	Dupliquer
Arborrescence :	marque d'instrument	Annuler
Nom :	Instrument de percussion	
Date de création :	2012-05-21 15:39:02	
Date limite de traduction :	2012-05-24 15:39:12	
Date limite de relecture :	2012-05-23 15:39:08	
Type de traduction :	valeur	
Langue de référence :	français	
Langue de traduction :	english	
Traducteur :	lapin	
Relecteur :	ours	

Avancement	
Traduite :	100%
Relue :	100%
Validée :	100%

Statut	Fiches
terminé	▶ WoodBrass
terminé	▶ Bergerault

<< < > >>

Fig. 3.9 : Interface de la page Project de l'application E-Translate

### 3.4.2. Le deuxième sprint

Le deuxième sprint est plus simple à mettre en place que le premier car la création de la structure de l'application est déjà faite. Lors du deuxième sprint la programmation a consisté à terminer la programmation de la gestion du workflow et à mettre en place un système d'arborescences.

Travail à réaliser	Développeur
Adaptation des interfaces au format de ceux du MediaContent	Jérémy
Amélioration des interfaces de gestion des projets	Cyrille
Création d'un système d'arborescences	Cyrille
Création de l'interface de gestion du profil de l'utilisateur	Jérémy
Mise en place des notifications	Jérémy

#### Création d'un système d'arborescences

Lors de la création d'un projet, un administrateur doit pouvoir sélectionner des éléments d'un arborescence afin de pouvoir créer par la suite les fiches. C'est pourquoi il est nécessaire de mettre en place un système de visualisation de donnée sous forme d'arborescences. Malheureusement PRADO ne propose pas cette fonctionnalité, il va donc falloir en créer une.

Avant de commencer la programmation des classes de l'arbre il faut d'abord s'intéresser aux données qui seront utilisées pour le créer. Un arbre pourra être créé à partir de deux types de donnée différentes : soit à partir d'une base de donnée (pour l'arborescence d'un projet), soit à partir d'un fichier xml (pour l'arborescence complète).

Il va aussi falloir s'intéresser aux différentes actions qui pourront être faites sur un arbre. Par exemple, lors de la création d'un projet, les fiches de celui-ci seront choisies à partir d'un arbre. Cela signifie que les nœuds de ces arbres pourront être sélectionnées.



Maintenant que l'on sait dans quelle direction partir, nous pouvons passer à la programmation. Voici une représentation des classes écrites en PHP décrivant le système d'arborescences :

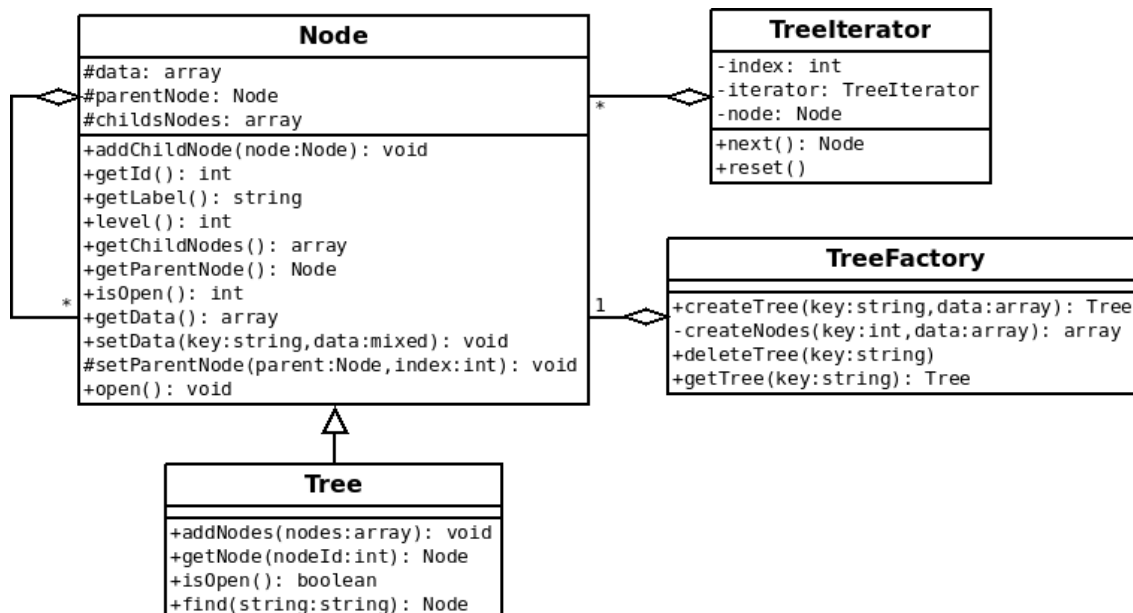


Fig. 3.10 : Diagrammes de classe de l'arborescence

Comme le montre la figure 3.10, la création d'un système d'arborescences ne va nécessiter que quatre classes appelées **Node**, **Tree**, **TreeIterator** et **TreeFactory**.

La classe **Node** est sans doute la classe la plus importante de l'arborescence. C'est uniquement celle-ci qui va mettre en place l'arborescence. Les autres classes ne servent qu'à ajouter des fonctionnalités ou à simplifier la création de ses instances. Les instances de cette classe représentent les nœuds de l'arbre. Un nœud connaît l'ensemble de ses fils ainsi que son père. Il possède également d'autres informations comme un identifiant, un label, un niveau de profondeur (dans l'arbre), etc.. Ces informations sont stockées dans un tableau permettant ainsi de pouvoir en ajouter d'autres si nécessaire.

La classe **Tree** va hériter de la classe **Node**. Cette classe correspond à un type de nœud particulier car il ne peut pas posséder de parent. Il ajoute également d'autres fonctionnalités aux nœuds classiques comme par exemple la possibilité de récupérer un nœud par son identifiant ou encore en rechercher un à partir d'une chaîne de caractère.

La classe **TreeIterator** permet de parcourir un arbre. Cela signifie que cette classe offre la possibilité de consulter un arbre nœud par nœud. La méthode `reset()` permet d'initialiser le **TreeIterator** au début de l'arbre et la méthode `next()` permet de passer au nœud suivant. Cette classe peut donc être utilisée pour effectuer des actions sur chacun des nœuds d'un arbre.

La classe **TreeFactory** permet de simplifier la création d'une arborescence. La création d'un arbre nœud par nœud peut être une étape longue et fastidieuse. L'utilisation de classe permettrait de pouvoir créer d'un seul coup toute l'arborescence grâce à la méthode `createTree()`. Il suffit pour cela de lui passer en paramètre un tableau contenant les données des nœuds à créer. Cette classe est donc utile pour créer un arbre à partir d'une base de données. Cependant, pour la création d'un arbre à partir d'un fichier xml, il peut être plus intéressant de créer une autre classe spécialisée mais cette étape n'est pas prévue pour ce sprint.



### 3.4.3. Le troisième sprint

Ce dernier sprint sera plus court que les autres étant donné qu'il va falloir également former un autre développeur de l'équipe de développement afin que la suite de l'application puisse être reprise. Ce sprint va donc consister à l'amélioration de quelques fonctionnalités de l'application ainsi que l'écriture de script d'automatisation de l'installation de certaine partie de l'application.

Travail à réaliser	Développeur
Amélioration du système de notification	Jérémy
Gestion de la traduction des notifications et des statuts des fiches	Jérémy
Avancement de l'interface d'ajout des fiches à un projet	Cyrille
Installation de l'application sur l'ordinateur du nouveau développeur de l'application	Jérémy
Explication du fonctionnement de l'application au nouveau développeur	Cyrille
Écriture d'un script de déploiement automatique d'un workflow	Jérémy
Écriture d'un script de création automatique d'un déploiement	Cyrille

#### Avancement de l'interface d'ajout des fiches à un projet

L'interface qui utilisera le plus le système d'arborescence de l'application est l'interface qui permettra d'ajouter des fiches lors de la création d'un projet. Dans cette page, l'arbre sera créé à partir d'un fichier xml récupéré depuis le MediaContent afin de pouvoir ensuite sélectionner des nœuds qui concerneront les fiches à ajouter. Cependant, dans la version actuelle de cette interface, les arbres ne sont pas encore générés par fichier xml.

Voici une représentation de l'interface de cette page :

Fig. 3.11 : Interface de la page d'ajout des fiches à un projet



Comme le montre cette interface, le système d'arborescence qui a été mis en place lors du sprint précédent a été utilisé. En ce qui concerne l'affichage de l'arbre, il a simplement été utilisé un élément du framework PRADO qui permet de répéter la mise en forme d'un nœud. Cela signifie qu'il suffit seulement de parcourir l'arbre en utilisant la classe `Treeliterator` et de mettre en forme les données de chacun des nœuds pour ensuite les afficher.

Pour dérouler une branche d'un arbre, il suffit de cliquer sur l'icone en forme de triangle à gauche d'un nœud. Cependant, en utilisant les éléments classiques de PRADO, l'ouverture d'un nœud nécessite de recharger la totalité de la page. Pour optimiser le temps d'ouverture d'un nœud, il peut donc être plus intéressant de ne recharger que les nœuds qui ne sont pas déjà affichés. Pour cela, il suffira d'utiliser des éléments spécial de PRADO qui utilisent le langage Ajax. L'Ajx permet d'effectuer des requêtes au serveur sans avoir à recharger la page dans le navigateur. Cela signifie qu'avec cet outil, un nœud pourra être déroulé sans avoir à toucher aux autres nœuds de l'arbre.

De la même manière que l'ouverture des nœuds, le fait de cocher des cases des nœuds va envoyer une requête au serveur afin de mettre à jour en temps réel l'arbre chargé en mémoire sur le serveur. Cela permet ainsi d'éviter que toutes les cases ne se décochent lorsque l'on recharge la page dans le navigateur.

La dernière chose qui a été mise en place dans cette interface est l'ajout d'informations supplémentaires sur un nœud lorsque la souris passe devant. Ces informations vont prendre la forme de drapeau indiquant si le nœud existe déjà dans d'autre projet. Le drapeau représente la langue de traduction du projet. Ainsi un administrateur pourra savoir si le nœud est déjà en cours d'utilisation ou non. De plus, si le nœud est déjà utilisé par un projet dont la langue de traduction est la même, le nœud ne pourra pas être sélectionné. Pour éviter de surcharger l'affichage d'un arbre, ces informations seront automatiquement cachées lorsque la souris n'est plus devant le nœud.

En ce qui concerne le système de recherche dans l'arbre, les boutons sont déjà présents mais ils ne sont pas encore opérationnels. La gestion de la recherche n'était pas prévu pour ce sprint.



## BILAN TECHNIQUE

Le projet concerné par ce rapport consistait à la création d'une application web intégrée sous la forme d'un module dans l'application MediaContent de la société Compilsoft. Elle est utilisée pour gérer la traduction des données du MediaContent dans divers langues.

Une ancienne version de cette application existait déjà mais ne satisfaisait pas l'équipe de développement de Compilsoft. C'est pourquoi mon rôle a été de recréer totalement cette application à partir de zéro. Cependant la nouvelle version de cette application devait également intégrer un nouvel outil nommé Activiti encore inconnu à l'entreprise.

Le stage a d'abord commencé par l'analyse de l'ancienne application E-Translate afin d'avoir quelques bases sur la méthode de développement de l'entreprise. Cette analyse s'est faite sans trop de problèmes étant donné que le framework PRADO est assez facile à comprendre. La plus grosse difficulté a été la compréhension de la base de donnée car les tables étaient nombreuses et la méthode de stockage des données n'était pas forcément intuitif.

Il a ensuite fallu étudier la plateforme Activiti dont l'utilité est de gérer les workflows. Cet outil n'a encore jamais été utilisé dans la société Compilsoft. C'est pourquoi son apprentissage n'a pas été aussi facile qu'avec le framework PRADO étant donné que les autres développeurs ne pouvaient pas m'aider. De plus, cet outil n'est apparu que très récemment, ce qui signifie qu'il n'est pas encore très documenté sur Internet. Mais après deux semaines intensives Jérémy DENIS et moi même avons au final réussi à créer une petite application de démonstration de cet outil pour la société Compilsoft.

Le stage a ensuite continué par l'écriture du cahier des charges de la nouvelle application E-Translate. Cette étape était la plus importante du stage car le cahier des charges devait rassembler tout ce qui allait concerner le développement de l'application par la suite. Durant cette étape, il a fallu beaucoup discuter avec certains membres de l'équipe de développement afin de clarifier tout ce que l'application devait pouvoir faire. Le cahier des charges a beaucoup évolué avant d'établir la version finale

La dernière partie du stage concernait la programmation de l'application. Étant donné que lorsque cette partie a commencé il ne restait plus que 3 semaines de stage, il était évident que l'application ne pouvait pas être finie dans les temps. C'est pourquoi la programmation de l'application a essentiellement tourné autour du développement de la gestion du workflow afin que les développeurs qui prendront la suite du développement puissent comprendre comment utiliser Activiti.

Pour conclure ce bilan, on peut dire que bien que l'application E-Translate ne soit pas entièrement terminée, mon stage aura permis de faire découvrir à l'entreprise un outil puissant pour gérer leurs workflows. Pour ce qui est de l'E-Translate, le cahier des charges ainsi que les nombreux autres documents concernant le développement de l'application permettront aux développeurs qui prendront la suite de son développement de pouvoir travailler facilement.



## CONCLUSION

Ce stage aura été avant tout pour moi une première expérience dans le monde professionnel. J'ai ainsi pu découvrir comment était organisée la façon de travailler dans une entreprise. J'ai également pu apprendre des méthodes de travail comme par exemple la méthode Scrum qui est utilisée dans certaines grandes entreprises. Cela m'a donc permis de voir une autre façon de travailler que celle que l'on peut voir à l'école.

En ce qui concerne le travail en lui-même, j'ai été très surpris par l'utilisation que l'on pouvait faire du langage PHP. En cours j'ai été habitué à utiliser le PHP de façon un peu brute, c'est à dire sans utiliser la programmation orienté objet. L'utilisation du framework PRADO m'a ainsi fait découvrir une autre facette de ce langage qui m'était inconnue.

J'ai également appris durant ce stage qu'il fallait toujours faire preuve de beaucoup de rigueur dans son travail. Cette rigueur va permettre d'améliorer la compréhension de son travail par nous mais surtout par les autres développeurs qui travaillent dans la même équipe.

Cette expérience m'aura donc appris beaucoup de chose que ce soit au niveau du travail ou de la façon de travailler. Elle m'aura également permis de changer ma vision de la programmation web.





## SUMMARY

My training session took place in the company Compilsoft. This Company has a content management application named MediaContent which is composed of several modules. My role was to completely rewrite the module named E-Translate which enables to translate data of this application.

The old version of this module didn't match with the needs of the development team of Compilsoft. So I was asked to make a new module in order to solve the problems of the old E-Translate. The new module will allow us to integrate tools which were not use by the Compilsoft's developers. This tools named Activiti enables to manage workflows. It means that the programming of the module consisted only of the creation of interfaces for the workflow's tasks.

The training started by the study of the older E-translate. It made me understand the way to program and also to discover the tools used by the developement team of compilsoft. It also made me to study the different features of the original module.

Before the start of the creation of the new module, I interested myself in activiti in order to make a presentation for the compagny. This part of the training consisted on the creation of a slideshow, and a little application which demonstrate how it works.

The next part of my training was the realization of the specifications of the new E-Translate. It was the most important part of the training because it regroups all the features the new module will bring.

The last part of my training consisted to create the module, with the Scrum Method. To realise the module, I split up the work in three sprints. In each sprint, the most important work was to develop the interfaces for the workflow's tasks.

The application is not over yet after the period of my training session, so other developers will continue to work on this project.



## BIBLIOGRAPHIE

- Le site officiel de PRADO : <http://www.pradosoft.com/>
- Le site officiel d'Activiti : <http://activiti.org/>
- Documentation sur le CSS : <http://www.w3schools.com>
- Manuel PHP : <http://php.net/manual/fr/index.php>



## LEXIQUE

<b>Activiti</b>	Activiti est une plateforme destinée aux développeurs, aux administrateurs système et aux gestionnaires permettant de gérer des workflows et des BPM (Business Process Management)
<b>BPMN</b>	Business Process Modeling Notation (BPMN) est une notation graphique standardisée pour modéliser des procédures d'entreprises dans un workflow.
<b>HTML</b>	L'Hypertext Markup Language (HTML) est le format de donnée conçu pour représenter les pages web. C'est un langage de balisage qui permet d'écrire de l'hypertexte.
<b>Javascript</b>	JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web interactives mais aussi côté serveur.
<b>JSON</b>	JSON (JavaScript Object Notation) est un format de données textuel et générique. Il permet de représenter de l'information structurée.
<b>Maven</b>	Apache Maven est un outil logiciel libre pour la gestion et l'automatisation de production des projets logiciels Java en général. L'objectif recherché est de produire un logiciel à partir de ses sources, en optimisant les tâches réalisées à cette fin et en garantissant le bon ordre de fabrication.
<b>PHP</b>	Le PHP (Hypertext Preprocessor) est un langage de scripts libre4 principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande. PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes.
<b>PRADO</b>	PRADO (PHP Rapid Application Development Object-oriented) est un framework libre écrit en PHP 5. Il permet de créer des applications web en utilisant le principe du pattern Modèle-Vue-Contrôleur. Cela signifie qu'il sépare la programmation de l'interface de la programmation de contrôle dans deux fichiers.



**Produit multicanal** Un produit multicanal est un produit qui utilise une stratégie de distribution qui recourt à plusieurs canaux (physiques et/ou virtuels) de distribution en parallèle. L'objectif d'une stratégie multicanal est d'aboutir à la combinaison la plus synergique possible.

**Scrum** Scrum est une méthode agile dédiée à la gestion de projets. La méthode s'appuie sur le découpage d'un projet en incréments, nommés "sprint", ainsi que l'auto-organisation de l'équipe de développement.

**Workflow** Un workflow (anglicisme) est la représentation d'une suite de tâches ou opérations effectuées par une personne, un groupe de personnes, un organisme, etc. Le terme flow renvoie au passage du produit, du document, de l'information, etc., d'une étape à l'autre.

**XML** Le XML (Extensible Markup Language) est un langage informatique de balisage générique.

Source : [fr.wikipedia.org](http://fr.wikipedia.org)



## ANNEXES

<b>Annexe 1. Activiti .....</b>	<b>46</b>
Introduction .....	46
1. Qu'est-ce qu'Activiti ? .....	46
1.1. A quoi sert-il ? .....	46
1.2. Historique d'Activiti .....	46
1.3 Architecture d'Activiti .....	47
2. Fonctionnement d'Activiti .....	49
2.1 Description de BPMN .....	49
2.2 Le cycle de vie d'un processus. ....	51
3. L'intérêt de l'utilisation d'Activiti .....	52
Conclusion .....	53
<b>Annexe 2. Chronologie du travail .....</b>	<b>54</b>
<b>Annexe 3. Explication technique du workflow .....</b>	<b>60</b>
Les variables de démarrage du processus .....	61
Description des tâches .....	61
Description des événements intermédiaires .....	62



## Annexe 1. Activiti

### Introduction

Activiti est une plateforme récente de modélisation de workflow. Dans le cadre du stage que nous sommes en train d'effectuer dans l'entreprise Compilsoft, il nous a été demandé de maîtriser cette technologie afin de déterminer s'il serait possible de l'utiliser dans le futur projet d'implémentation de l'e-translate que nous devons réaliser. Pour cela nous allons développer cette question en trois parties. Dans un premier temps nous verrons ce qu'est Activiti, de quoi est composé cette technologie. Puis, nous verrons comment fonctionne Activiti. Enfin nous étudierons ce qui fait ou non l'intérêt de cette technologie.

### 1. Qu'est-ce qu'Activiti ?

Avant d'entrer dans une explication détaillée sur le fonctionnement d'Activiti, il faut d'abord comprendre quel est son but et également son utilité au sein d'une entreprise.

#### 1.1. A quoi sert-il ?

Activiti est une plateforme destinée aux développeurs, aux administrateurs système et aux gestionnaires permettant de gérer des workflows et des BPM (Business Process Management). Cela signifie que cette plateforme est destinée aux entreprises qui souhaiteraient mettre en place un système de gestion structuré des différentes activités de leur entreprise. L'administration de ces activités deviendrait ainsi beaucoup plus simple car Activiti met à disposition beaucoup d'outils pour traiter les activités.

Activiti va donc offrir la possibilité de représenter une activité d'entreprise de façon claire et facile à comprendre pour un non spécialiste. Cette activité appelée « processus » va prendre la forme d'un diagramme représentant la circulation d'une ressource entre les différents postes de travail de l'entreprise.

Mais l'utilisation principale d'Activiti réside dans l'exécution de ces processus. Cela signifie qu'Activiti est capable de gérer toute la chaîne de traitement d'une ressource grâce à une machine virtuelle à laquelle les utilisateurs pourront accéder à partir d'une interface.

#### 1.2. Historique d'Activiti

C'est en 2010 que commence le projet Activiti, démarré à l'initiative de Tom Baeyens et Joram Barrez le fondateur et le développeur principal de JBPM une autre plate-forme de modélisation de workflow. Activiti a aussi été fondé par Alfresco Software une entreprise connue pour avoir développé un logiciel de gestion de contenu (ECM) du même nom.



C'est un projet open source sous licence Apache dont le principal but est de fournir une plate-forme solide capable de fonctionner sous de nombreux environnements et de manière autonome, c'est à dire sans l'aide d'une technologie extérieur et implémentant le nouveau standard BPMN 2.0 pour la modélisation de workflow.

C'est en décembre 2010 que sort la première version stable d'Activiti la version 5.0. Activiti n'a pas eu de versions de 1 à 4 car elle reprend les principaux outils existant dans une autre plate-forme de modélisations de workflow JBPM. JBPM est actuellement en version 5 et continue son développement en parallèle d'Activiti ces deux plate-formes de modélisation de contenu étant les premières à implémenter le nouveau standard BPMN 2.0.

Le langage de notation pour modéliser un workflow est le Business Process Modeling Notation (BPMN). Il a été créé en 2004 et est actuellement en version 2.0. Ce n'est que depuis la version 5 de JBPM que des plate-formes sont capables de supporter et implémenter les nouveaux standards liés à BPMN 2.0. C'est l'OMG (Object Management Group) qui se charge de maintenir et développer cette notation. Alors que la première version de cette notation n'offrait que des symboles pour représenter des workflows simples, cette seconde version a été enrichie et intègre désormais des outils pour représenter un plus grand nombre d'éléments, de conditions et de symboles pour décrire les échanges pouvant avoir lieu entre les différentes tâches d'un processus ainsi que les différents chemins que peut prendre un flux de données lors de l'exécution de ce workflow.

De plus cette nouvelle version de ce langage intègre une correspondance en XML pour une plus grande portabilité, pour que l'utilisation dans les programmes de ces langages soit plus facile pour les développeurs mais aussi pour que cette notation soit régie par des règles pour son intégration. Ainsi l'intégration de la notation BPMN dans Activiti se fait par la correspondance d'un diagramme utilisant cette notation et un fichier XML qui sera compris par Activiti.

### 1.3 Architecture d'Activiti

Activiti est une plate-forme entièrement développée en Java. Elle est composée de différents composants qui lui permettent d'assurer les services dont pourrait avoir besoin les développeurs. Activiti est ainsi composé de quatre grands composants :

#### Activiti Engine

Il s'agit du cœur de la plate-forme Activiti qui comprend différents outils pour exécuter des processus écrits grâce à la notation BPMN 2.0. avec une intégration des nouvelles possibilités offertes par le standard BPMN 2.0 comme : un enrichissement des événements disponibles lors de l'exécution d'un processus lié au workflow (timer et sous-processus), possibilité d'ajouter du contenu lié à une tâche, et de nouvelles règles de communication liées aux tâches via des gateways. Pour permettre une exécution des processus liés à un workflow ce composant intègre une « process virtual machine » qui correspond à la machine virtuelle qui met en place ce qui est décrit dans le fichier XML correspondant au workflow. C'est cette machine virtuelle qui gère le bon fonctionnement des processus d'un workflow.



### Activiti Modeler

Cet élément est un composant de modélisation rapide de workflow utilisant la notation BPMN 2.0. Il est capable grâce à des schémas BPMN 2.0 de nous générer le code XML associé pour qu'il puisse être déployé dans l'Activiti Engine et ainsi qu'on puisse observer son fonctionnement au cours du temps. Inversement, à partir du code XML définissant un workflow et décrivant sa structure, ce composant est capable de retrouver le diagramme XML associé pour en permettre une visualisation graphique permettant à l'utilisateur de se représenter le workflow sur lequel il travaille.

### Activiti Explorer

Cette application correspond à une interface graphique de l'activiti engine. On peut y voir l'état d'un workflow lors de son exécution et aussi gérer les différentes instances des processus fonctionnant sur le serveur. C'est aussi ce composant qui propose une vue graphique de la base de donnée d'Activiti. Il est possible d'y réaliser de nombreuses actions comme de déployer de nouveaux diagrammes ou même de gérer les différents processus en cours d'exécution.

### Activiti Cycle

Il s'agit d'un composant pour mettre en commun l'analyse, le développement et le déploiement d'une ressource. Il propose une interface où différentes personnes peuvent travailler dessus en commun tout en observant où en est le projet.

La plateforme Activiti est composée d'une base de données contenant les différentes informations requises au traitement des différents processus. Cette base de donnée est nativement de type H2 c'est à dire qu'elle est chargée en mémoire lors de l'utilisation de l'Activiti engine. Ainsi cette base de données possède des tables temporaires qui se vident après leur utilisations., ceci pour avoir une plus grande rapidité lors de interrogation de la base de données. Pourtant pour un meilleur interfaçage avec certains langages et pour permettre une plus grande utilisations de la plateforme, il est possible de convertir cette base de données au format MySQL.

Dans Activiti, il a aussi été réalisé une implémentation de l'API REST, ce qui permet à un utilisateur de réaliser différentes requêtes pour interroger la base de données ou réaliser différentes actions avec l'Activiti Engine comme par exemple démarrer un processus ou terminer une tâche. Cela permet de réaliser les principales actions qu'un utilisateur voudrait réaliser de manière simple. La réponse aux différentes requêtes réalisées en utilisant cette API nous est fournie au format JSON afin de mieux structurer les données et obtenir une réponse facilement traitable avec des langages tel que le Java, le JavaScript ou même le PHP.



## 2. Fonctionnement d'Activiti

Nous allons maintenant nous intéresser plus en détail sur le fonctionnement d'Activiti. Pour pouvoir utiliser Activiti il faut d'abord comprendre comment réaliser un diagramme représentant une activité d'entreprise. Pour cela, Activiti implémente la notation standardisée de modélisation des procédures d'entreprise (BPMN) afin de rendre les diagrammes compréhensibles depuis l'analyste jusqu'à l'utilisateur.

### 2.1 Description de BPMN

Les différents éléments des diagrammes de BPMN ont beaucoup changé au cours du temps. Nous aborderons ici que certains éléments de BPMN 2.0 qui représentent ceux utilisés par Activiti.

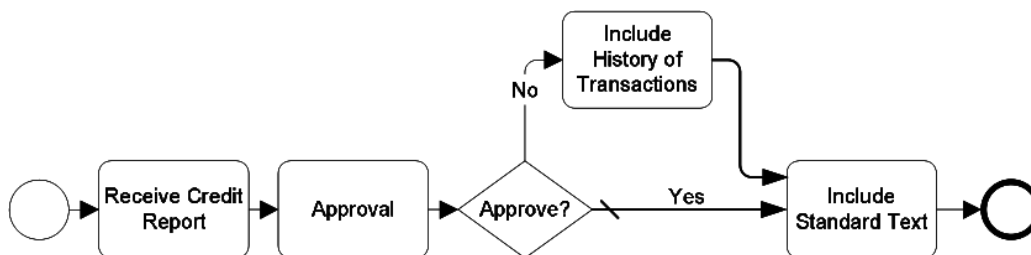


Fig. 2.1 : exemple de diagramme BPMN.

Ces éléments peuvent prendre 4 formes différentes :

- les « tasks » : Une tâche correspond à un travail qui sera réalisé de façon atomique. Ce traitement pourra par exemple consister en la modification de données contenues dans une base de données. Il existe plusieurs types de tâche en fonction du travail à réaliser :
  - Les « user tasks » : Cela signifie que le travail va être réalisé par un utilisateur. Elle ne se terminera qu'une fois qu'il aura complété son travail.
  - Les « services tasks » : Ce type de tâche correspond à un travail réalisé de façon automatique par une classe JAVA.
  - Les « script tasks » : Ce type de tâche correspond à une courte action réalisée de façon automatique mais ayant la possibilité d'utiliser des langages différents du JAVA comme par exemple le Ruby, le JavaScript, le Python...

Task



Task



Task



Task



- les « events » : Un événement correspond à quelque chose qui peut se produire durant l'exécution d'un processus. Il peut prendre la forme d'un événement de démarrage, intermédiaire ou final.

**Start**



**Intermediate**



**End**

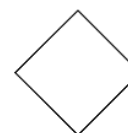


Il existe plusieurs types d'événements :

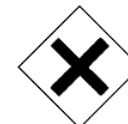
- Les événements vides : Ils correspondent aux événements de début et de fin d'un processus.
- Les messages : Ces événements se produiront lorsqu'il y a une réception ou un envoi d'un message.
- Les erreurs : Un événement de type erreur peut être déclenché lorsqu'une tâche ne s'est pas terminée correctement.
- Les timers : Ces événements se produiront lorsqu'un certain temps se sera écoulé.



- Les « gateways » : Un gateway est utilisé en sortie d'une tâche afin d'intégrer des opérateurs logiques permettant ainsi de cibler plusieurs tâches différentes. Cela signifie que la circulation des ressources entre les tâches sera différente en fonction de la validation des conditions des gateways. Les principaux types de gateway utilisés par Activiti sont :



- Le OU exclusif : Ce type de gateway permet d'offrir plusieurs chemins possibles lorsqu'une tâche a terminée son travail. Le choix pourra être déterminé à partir de condition.
- Le OU logique : Pour celui là, le chemin emprunté sera encore calculé en fonction des conditions mais il autorise également la possibilité de prendre plusieurs chemin en même temps. Il pourra donc y avoir plusieurs tâches qui seront exécutées en même temps.
- Le ET logique : Ce dernier permet d'exécuter plusieurs chemins en parallèle sans conditions.



Ces diagrammes pourront être réalisés à partir de l'interface graphique d'Activiti Designer, mais également à partir d'un fichier XML respectant le format mis en place par BPMN.

## 2.2 Le cycle de vie d'un processus.

Maintenant que l'on sait ce qu'est un diagramme BPMN, nous allons nous intéresser à l'exécution de ce diagramme au sein d'Activiti. Pour cela, nous allons prendre un exemple simple de diagramme afin de comprendre les différents mécanismes qui sont mis en place par Activiti. Ce diagramme correspond à une demande de vacance.

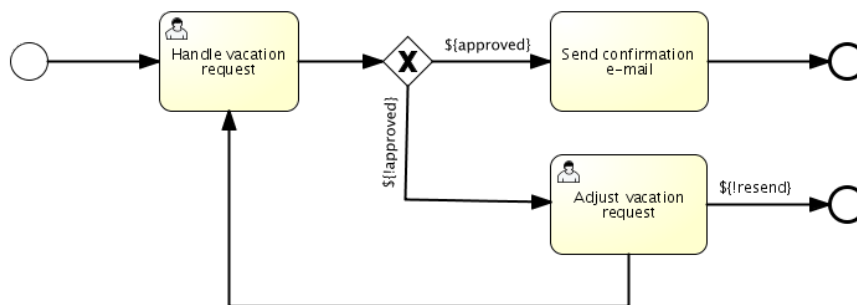


Fig. 2.2: Exemple de diagramme BPMN (demande de vacance)

La première chose à faire est d'intégrer ce diagramme à la machine virtuelle d'Activiti. Pour cela, il suffit de « déployer » le fichier XML ou une archive zip contenant les fichiers XML s'il y en a plusieurs.

Un utilisateur va ensuite pouvoir démarrer le processus. Pour cet exemple, l'utilisateur va le démarrer en spécifiant 3 variables à l'événement initial :

- numberOfDays, correspondant au nombre de jours de vacances demandés,
- startDate, correspondant à la date de commencement des vacances,
- vacationMotivation, correspondant à la raison pour laquelle l'employé souhaite prendre des vacances.

Les variables ne peuvent être liées qu'à des tâches ou au commencement d'un processus comme ici. Ce sont elles qui vont contenir les informations sur lesquelles les tâches vont pouvoir travailler. Elles sont donc nécessaires à tout processus.

Après que le processus soit démarré, le workflow va arriver à une tâche nommée « Handle vacation request ». Cette tâche correspond à une tâche de type « User Task », cela signifie que c'est un utilisateur qui va réceptionner la demande de vacance. On peut alors se demander à quel utilisateur va être affectée cette tâche ? Dans Activiti, une tâche peut être affectée soit à un utilisateur, soit à un groupe. Si la tâche est affectée à un utilisateur, seul cet utilisateur pourra accéder à cette tâche. Si la tâche est affectée à un groupe, cette fois-ci tous les utilisateurs appartenant au groupe pourront récupérer cette tâche. Cependant, une fois qu'un des utilisateurs du groupe aura récupéré la tâche, les autres ne pourront plus la voir.

L'utilisateur va ensuite pouvoir effectuer les traitements adéquates avant de terminer la tâche. Dans cet exemple, les traitements vont simplement consister en la validation ou le refus de la demande de vacance. Il va donc mettre à 'true' ou à 'false' la variable « vacationApproved » affectée à la tâche.

Une fois que cette tâche est terminée, le workflow va arriver à un gateway de type exclusif. Cela signifie que la suite du parcours va différer en fonction de la condition associée. Si « vacationApproved » est à 'true', la suite du workflow va arriver à une tâche permettant d'envoyer un mail puis le processus va ensuite se



terminer. Si la variable est à 'false', le workflow va retourner sur un « user task ». Cette fois-ci la tâche est affectée à l'utilisateur qui a démarré le processus. Il va pouvoir choisir entre ré-envoyer sa demande avec une nouvelle motivation à la tâche précédente ou alors décider d'arrêter la demande de vacance et donc terminer le processus.

Cet exemple illustre parfaitement la gestion d'une procédure d'entreprise dans son intégralité, c'est à dire de la conception du workflow jusqu'à son exécution. Activiti a donc l'avantage de nous permettre de se débarrasser de la programmation du workflow et également de la gestion des droits des utilisateurs pour se consacrer uniquement à la conception de l'interface de manipulation du workflow.

### 3. L'intérêt de l'utilisation d'Activiti

Activiti fonctionne de la manière suivante : on a une application que l'on doit déployer sur un serveur d'application (comme Tomcat par exemple). Ensuite on peut accéder aux différents services proposés par Activiti comme par exemple l'Api Rest ou l'Activiti explorer. Activiti est une plateforme de modélisation de workflow qui lors de la réalisation de notre première application nous est apparue comme comportant de nombreux points positifs mais aussi quelques points négatifs.

Pour ce qui est de l'aspect positif de cette technologie, Activiti permet de réaliser des applications structurées où le cycle de vie d'un processus est bien représenté. En effet, Activiti permet d'avoir une vision de notre processus où l'on distingue où se trouvent les différentes tâches à réaliser, où commence le processus et le chemin que prendra le flux de données lors de son exécution. Grâce à l'Activiti modeler il est très facile d'avoir le code XML correspondant au diagramme voulu sans avoir à retenir les balises propre à cette notation.

Enfin le point le plus important d'Activiti est que cette technologie offre une grande malléabilité dans la structure finale d'une application. En effet grâce à cette technologie il est possible de réaliser une application complètement personnalisable en fonction des besoins auxquels nous devons répondre. En effet après avoir préalablement créé les tâches faisant parties de notre workflow, il est possible de les disposer comme on veut, de les faire exécuter selon un nombre de fois voulu et même sous une contrainte de temps.

Néanmoins, cette technologie comporte aussi quelques point négatif, à commencer par le fait qu'il s'agit d'une technologie assez jeune qui va sûrement évoluer dans les années qui viennent. Cette plateforme n'est pas très intuitive à utiliser et l'on rencontre quelques difficultés lors de la prise en main. De plus le fait que cette technologie soit entièrement tournée vers le Java possède certes l'avantage d'être portable mais limite les actions que l'on peut réaliser dessus. Enfin la communauté autour de cette technologie n'est pas encore beaucoup développée malgré le fait que l'on puisse observer sur les forums qu'il y a une communauté assez active bien qu'en trop petit nombre, on peut notamment lire dans de nombreux articles et forum des avis de personnes conseillant Activiti par rapport à d'autre plateforme de modélisation de workflow comme JBPM ou Bonita.



## Conclusion

Pour résumer, on peut dire qu'Activiti possède toute une palette d'outils permettant de pouvoir créer des diagrammes BPMN claires et compréhensibles, de pouvoir ensuite les déployer au sein d'un moteur de gestion des processus et de proposer une interface claire montrant le cycle de vie d'un processus. Il offre aussi la possibilité de pouvoir créer d'autres interfaces en utilisant le langage Java mais également d'autres langages tournées vers le Web.

Il est donc évident que cette plate-forme peut avoir une grande utilité au sein d'une entreprise comme Compilsoft dont la principale activité est de construire des interfaces Web pour les entreprises permettant de manipuler leurs données. Activiti permettrait ainsi d'accélérer la création de ces interfaces en s'abstrayant de la programmation de la gestion des workflows.



## Annexe 2. Chronologie du travail

Voici une présentation chronologique du travail effectué au cours de ces 10 semaines de stage. Les diagrammes suivant contiendront les différentes tâches accomplies par Jérémie DENIS et par moi-même, le deuxième stagiaire de l'IUT avec qui je travaillais.

Jour	Lundi								Mardi								Mercredi								Jeudi								Vendredi							
Heure	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
Semaine 1	Installation des postes								Prise en main de l'architecture du MediaContent																															
Semaine 2	Férié								Création d'une application (PRADO)								Analyse de l'existant																							
Semaine 3									Étude d'Activiti																															
Semaine 4									Création d'une application avec Activiti																															
Semaine 5									Férié								Rédaction d'une présentation d'Activiti																							
Semaine 6									Férié								Rédaction du cahier des charges																							
Semaine 7	Scrum								Spécifications techniques								Férié																							
Semaine 8									Application E-Translate (sprint 1)																															
Semaine 9	Férié								Application E-Translate (sprint 2)																															
Semaine 10									Application E-Translate (sprint 3)																															

### Installation des postes

Lorsque nous sommes arrivés dans l'entreprise, il a d'abord fallu réinstaller convenablement les ordinateurs sur lesquels nous allions travailler.

Heure	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Jérémy	T1	T3		T3				T4								
Cyrille		T2														

- T1 : Enregistrement des données des anciens utilisateurs des ordinateurs sur lesquels nous avons travaillé.
- T2 : Installation de Windows 7 sur l'ordinateur utilisé pour faire des tests.
- T3 : Formatage des ordinateurs et installation de Xubuntu 10.04
- T4 : Installation des programmes (Eclipse, Mysql-Workbench, Tomcat, ...)



## Prise en main de l'architecture du MediaContent

Lorsque les ordinateurs ont été opérationnels, nous avons récupéré les applications déjà réalisées par les autres développeurs de la société Compilsoft, afin de comprendre comment était structuré le code de ces applications, ainsi que pour les tester localement sur nos ordinateurs.

Heure	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
Jérémy	T1		T2	T3	T4	T4					T5														
Cyrille			T1		T3																				

- T1 : Configuration des plugins de l'application Eclipse
- T2 : Configuration du client SVN permettant de récupérer les applications de Compilsoft
- T3 : Récupération des modules du MediaContent y compris l'E-Translate
- T4 : Compilation et installation des modules
- T5 : Étude du fonctionnement du MediaContent ainsi que ses modules

## Création d'une application avec PRADO

Après avoir étudié l'écriture des applications, nous avons tenté de créer notre propre application utilisant le framework PRADO et de l'intégrer dans le MediaContent.

Heure	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Jérémy	T1		T2				T4								T5	
Cyrille			T3													

- T1 : Création de page en utilisant le système de PRADO
- T2 : Utilisation des templates pour générer automatiquement l'en-tête et le pied d'une page
- T3 : Communication avec une base de donnée
- T4 : Insertion de l'application dans le MediaContent sous la forme d'un module
- T5 : Utilisation du système de changement de langue du MediaContent

## Analyse de l'existant

Durant cette tâche, nous avons étudié en détail le fonctionnement de l'ancienne application E-Translate afin de déterminer précisément le travail que nous devons réaliser.

Heure	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Jérémy	T1				T2						T5					
Cyrille					T3	T4										

- T1 : Configuration de l'application E-Translate
- T2 : Analyse du fonctionnement de l'application
- T3 : Configuration de la base de donnée afin de travailler tous les deux sur la même
- T4 : Analyse de la base de donnée
- T5 : Écriture d'un rapport nommé « Analyse de l'existant »



## Étude d'Activiti

Avant de commencer de travailler sur l'E-Translate, il a fallu étudier Activiti qui est un nouvel outil que la société Compilsoft voulait tester.

Heure	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Jérémy	T1								T2		T3	T4		T5				
Cyrille											T3		T4					

Heure	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Jérémy	T6										T8							
Cyrille	T7			T6				T8										

- T1 : Recherche de documentation sur Activiti et BPMN
- T2 : Installation d'Activiti
- T3 : Installation et configuration du plugins Activiti Designer dans Eclipse
- T4 : Configuration d'Activiti
- T5 : Réalisation de test avec Activiti-Explorer
- T6 : Étude du fonctionnement d'Activiti
- T7 : Test du plugins Activiti Designer
- T8 : Modification de la configuration d'Activiti pour intégrer une base de donnée MySQL

## Création d'une application avec Activiti

Notre chef de projet nous a demandé de réaliser une application de démonstration d'Activiti. Pour cette démonstration, nous avons choisi de programmer un jeu de question entre deux joueur en utilisant un workflow géré par Activiti.

Heure	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
Jérémy	T1		T3	T4	T5										T4					T6						
Cyrille			T2		T4															T8						

Heure	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
Jérémy	T6	T7				T11								T9								T9				
Cyrille	T8										T7			T10							T8					

- T1 : Analyse de l'application à créer
- T2 : Programmation d'un driver communiquant avec le web service d'Activiti
- T3 : Programmation d'un driver communiquant avec la base de donnée
- T4 : Programmation du driver globale
- T5 : Programmation de la gestion des droits des utilisateurs
- T6 : Programmation de la partie s'occupant de créer des utilisateurs
- T7 : Programmation de la partie s'occupant de créer une nouvelle partie dans le jeu
- T8 : Création du workflow et programmation des interfaces du jeu
- T9 : Intégration de l'application dans le MediaContent
- T10 : Programmation de l'interface de visualisation des résultats
- T11 : Programmation de l'interface de contrôle du jeu de l'administrateur





## Rédaction d'une présentation d'Activiti

En plus de l'application de démonstration d'Activiti, nous avons écrit un rapport et un diaporama afin de présenter et expliquer le fonctionnement d'Activiti.

Heure	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Jérémy	T1															T2								
Cyrille																								

- T1 : Écriture du rapport
- T2 : Écriture du diaporama

## Rédaction du cahier des charges

Lorsque l'on crée un nouveau projet, il faut d'abord commencer par écrire le cahier des charges

Heure	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Jérémy	T1						T2				T4				T5							
Cyrille																						

Heure	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44
Jérémy	T5										T6											
Cyrille																						

- T1 : Analyse du contenu que contiendra le document
- T2 : Écriture du contexte du projet
- T3 : Écriture de l'explication du workflow
- T4 : Écriture de l'explication des éléments de l'application
- T5 : Description des fonctionnalités de l'application
- T6 : Correction et modification du cahier des charges

## Scrum

Pendant 2 journées, l'équipe de développement a effectué une formation sur la méthode de travail en entreprise appelé Scrum. Jérémy et moi avons été invités à participer à la première journée de formation.

## Spécifications techniques

Le cahier des charges est un document qui doit être lisible par une personne non spécialisée. Le document qui contiendra des explications un peu plus technique s'appelle « Les spécifications techniques ».

Heure	1	2	3	4	5	6	7	8	9	10	11	12
Jérémy	T1				T3							
Cyrille												

- T1 : Analyse de la base de donnée
- T2 : Explication technique du workflow
- T3 : Explication du fonctionnement de l'application



### Application E-Translate (sprint 1)

La création de l'application a été divisée en 3 sprints. Le premier sprint va concerner la programmation des interfaces qui utiliseront le plus la communication avec Activiti.

Heure	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Jérémy	T1				T3					T5										
Cyrille					T2								T7							

Heure	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Jérémy	T5													T6						
Cyrille	T4																			

- T1 : Élaboration du premier sprint
- T2 : Création de la base de donnée
- T3 : Création des drivers
- T4 : Programmation des interfaces de gestion d'un projet
- T5 : Programmation des interfaces de travail sur une fiche
- T6 : Gestion des droits
- T7 : Création du workflow

### Application E-Translate (sprint 2)

Lors du deuxième sprint la programmation a consisté à terminer la programmation de la gestion du workflow et à mettre en place un système d'arborescences.

Heure	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Jérémy	T1		T2					T5					T6			
Cyrille			T3				T4									

Heure	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	
Jérémy	T7										T8					T9	
Cyrille	T4					T3			T4								

- T1 : Élaboration du deuxième sprint
- T2 : Adaptation des interfaces au format de ceux du MediaContent
- T3 : Amélioration des interfaces de gestion des projets
- T4 : Création d'un système d'arborescences
- T5 : Création de l'interface de gestion du profil de l'utilisateur
- T6 : Création du système de changement de langue
- T7 : Mise en place des notifications
- T8 : Correction des bugs de l'application
- T9 : Présentation du deuxième sprint



### Application E-Translate (sprint 3)

Ce dernier sprint va s'occuper de corriger les quelques bugs de l'application, et de continuer le développement de l'affichage des arbres, et l'amélioration de quelques fonctionnalités de l'application. Il va également concerner la formation d'un autre membre de l'équipe de développement afin qu'il puisse reprendre la suite de l'application.

Heure	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Jérémy	T1	T2								T3								T6		
Cyrille		T4														T5				

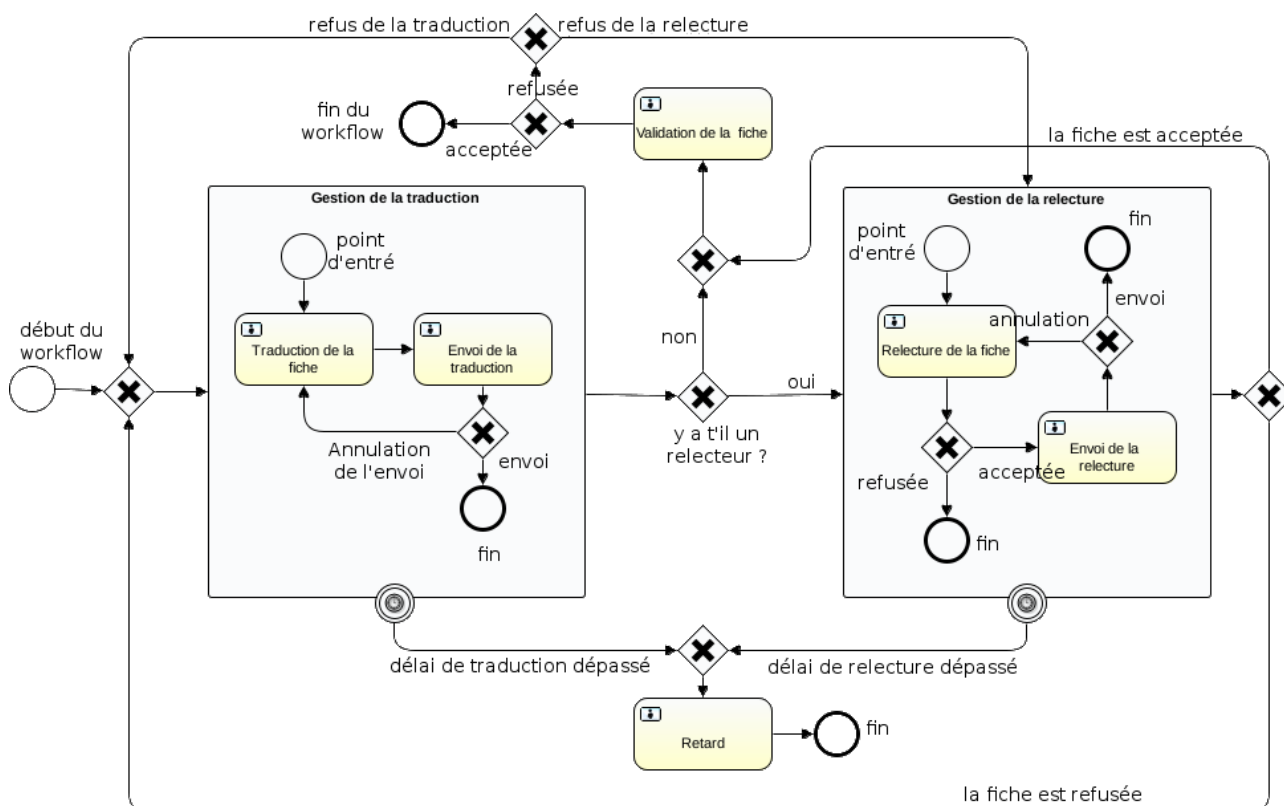
Heure	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Jérémy	T6			T5			T8										T10			
Cyrille	T10			T7								T9							T10	

- T1 : Élaboration du troisième sprint
- T2 : Amélioration du système de notification
- T3 : Gestion de la traduction des notifications et des statuts des fiches
- T4 : Avancement de l'interface d'ajout des fiches à un projet
- T5 : Correction de bugs de l'application
- T6 : Installation de l'application sur l'ordinateur du nouveau développeur de l'application
- T7 : Explication du fonctionnement de l'application au nouveau développeur
- T8 : Écriture d'un script de déploiement automatique d'un workflow
- T9 : Écriture d'un script de création automatique d'un déploiement
- T10 : Écriture de documents expliquant le fonctionnement de l'application

## Annexe 3. Explication technique du workflow

Pour pouvoir comprendre correctement les explications faites dans cette partie, il est important d'avoir quelque notions sur les diagrammes BPMN et Activiti en général.

Le workflow correspond au diagramme BPMN représentant la transition des états d'une fiche au sein de l'application E-Translate. Ces différents états représenteront par la suite les tâches des processus d'Activiti. Elles permettent de décrire l'évolution du travail effectué sur une fiche à traduire.



Ce workflow fait intervenir 2 utilisateurs (un pour la traduction et un pour la relecture) et le groupe correspondant aux administrateurs de l'application. Les tâches présentes dans le sous-processus « Gestion de la traduction » sont assignées au traducteur et les tâches du sous-processus « Gestion de la relecture » au relecteur. Quant aux autres tâches, elles sont cette fois-ci associées au groupe des administrateurs. Ce groupe et les utilisateurs de l'application sont donc présent dans Activiti afin de pouvoir interagir avec les processus.

Nous allons maintenant voir plus en détail les caractéristiques techniques du processus qui ne sont pas visible à partir du diagramme. Cela regroupe les données concernant les variables du processus, les tâches et les événements intermédiaires.



## Les variables de démarrage du processus

Ces variables sont globales à toutes les tâches. Cela signifie que les tâches peuvent accéder aux informations contenues dans ces variables. Il faut cependant qu'elles soient ajoutées à leur liste de variables.

Variables	Descriptions
translateDate	Date limite de traduction de la fiche.
reviewDate	Date limite de relecture de la fiche.
translator	L'identifiant Activiti de l'utilisateur qui s'occupe de la partie traduction.
reviewer	L'identifiant Activiti de l'utilisateur qui s'occupe de la partie relecture.
adminGroup	Le groupe Activiti contenant les administrateurs de l'application e-Translate
itemId	L'identifiant e-Translate de la fiche.

## Description des tâches

<b>Nom de la tâche :</b>	<b>Traduction de la fiche</b>
<b>Description :</b>	Cette tâche correspond à la traduction de la fiche. Au niveau de l'interface, le traducteur pourra remplir les différents champs avant de terminer ses modifications en cliquant sur le bouton « À envoyer ». Cette action termine ensuite la tâche.
<b>Est assignée à :</b>	\${translator}
<b>Variables :</b>	itemId (identifiant de la fiche)

<b>Nom de la tâche :</b>	<b>Envoi de la traduction</b>
<b>Description :</b>	Cette tâche concerne les tâches qui viennent d'être traduites. Elle offre la possibilité de pouvoir envoyer en groupe plusieurs fiches ou de renvoyer une fiche à la tâche de traduction si le traducteur pense avoir fait une erreur. La tâche se termine si l'une de ces deux actions a été faite.
<b>Est assignée à :</b>	\${translator}
<b>Variables :</b>	itemId (identifiant de la fiche) canceled (l'envoi est annulé)

<b>Nom de la tâche :</b>	<b>Relecture de la fiche</b>
<b>Description :</b>	Cette tâche est exécutée par le relecteur afin de pouvoir relire une fiche traduite. La tâche se termine lorsqu'il clique sur le bouton « Accepter » ou sur le bouton « Refuser ».
<b>Est assignée à :</b>	\${reviewer}
<b>Variables :</b>	itemId (identifiant de la fiche) accepted (la fiche est acceptée)



<b>Nom de la tâche :</b>	<b>Envoi de la relecture</b>
<b>Description :</b>	La tâche d'envoi de la relecture permet d'envoyer d'un seul bloc plusieurs fiches à la tâche de validation. Il est cependant possible d'annuler cet envoi en cliquant sur le bouton « Relire » qui renvoie la fiche à la tâche de relecture. La tâche se termine lorsqu'une de ces deux actions est effectuée.
<b>Est assignée à :</b>	\${reviewer}
<b>Variables :</b>	itemId (identifiant de la fiche) canceled (l'envoi est annulé)

<b>Nom de la tâche :</b>	<b>Retard</b>
<b>Description :</b>	Cette tâche permet à l'administrateur d'envoyer un mail aux utilisateurs ayant dépassé la date limite pour effectuer leur travail. Cette tâche se termine quand l'administrateur a envoyé le mail.
<b>Est assignée à :</b>	\${adminGroup}
<b>Variables :</b>	translator reviewer

<b>Nom de la tâche :</b>	<b>Validation de la fiche</b>
<b>Description :</b>	Cette tâche permet à l'administrateur de vérifier si la fiche est complétée. Il peut ensuite décider de la valider, de la renvoyer au traducteur ou de la renvoyer au relecteur. Cette action termine alors la tâche.
<b>Est assignée à :</b>	\${adminGroup}
<b>Variables :</b>	itemId (identifiant de la fiche) state (l'état peut être « validate », « toTranslator » ou « toReviewer »)

## Description des événements intermédiaires

<b>Événement du sous-processus « gestion de la traduction »</b>	
<b>Description :</b>	Cet événement se déclenche lorsque la date limite de traduction est dépassée. Il n'interrompt pas le sous-processus mais il démarre la tâche « Retard ».
<b>Type :</b>	time event
<b>Date :</b>	\${translateDate}

<b>Événement du sous-processus « gestion de la relecture »</b>	
<b>Description :</b>	Cet événement se déclenche lorsque la date limite de relecture est dépassée. Il n'interrompt pas le sous-processus mais il démarre la tâche « Retard ».
<b>Type :</b>	time event
<b>Date :</b>	\${reviewDate}