

Intégration d'un moteur de workflow dans le SI : Dématérialisation des Ordres de mission

Thierry Agueda

Direction des Systèmes d'Information UPMF
Université Pierre-Mendès France (Grenoble 2)
Bâtiment Langues et Nouvelles Technologies (LNT)
79 rue des universités
38100 GRENOBLE

Mathieu Panel

Direction des Systèmes d'Information UPMF
Université Pierre-Mendès France (Grenoble 2)
Bâtiment Langues et Nouvelles Technologies (LNT)
79 rue des universités
38100 GRENOBLE

Résumé

Dans le cadre de la simplification des procédures lancée au sein de notre établissement, nous avons mis en place un moteur de workflow afin de modéliser différents processus. Un workflow est une suite de tâches, humaines ou automatiques, réalisées par différents acteurs. Le processus de validation des ordres de mission a été le premier traité. Ce dernier se divise en plusieurs tâches auxquelles il faut associer les bons acteurs en fonction de l'initiateur du processus. Une application de référentiel de validation a été créée pour ce besoin. Ce référentiel permet d'instancier les processus avec les bons acteurs : qui valide qui ? Quel gestionnaire comptable est concerné ? Y a-t-il un valideur N+1 (chef de service ou de département) ? Les contraintes telles que la gestion des pays à risque ou empêcher une personne de se valider elle-même, sont intégrées dans le processus.

Mots-clefs

Dématérialisation, Workflow, Processus, Ordre de mission

1 Présentation et contexte

Le processus des ordres de mission au sein de l'Université se traduisait par le déplacement de paraphe entre les différents services et bâtiments. Son fonctionnement pouvait être différent d'une UFR à une autre. La présidence dans le cadre d'un projet de simplification des procédures, a établi un cahier des charges pour homogénéiser les pratiques. Après une étude des solutions existantes notre choix s'est porté sur l'intégration d'un moteur de workflow, plus polyvalent qu'une solution spécifique « ordre de mission », permettant de modéliser par la suite d'autres processus.

2 Concepts

2.1 Moteur de workflow Bonita

Bonita est une solution complète de gestion de processus. Nous utilisons la version OpenSource 5.9 (aujourd'hui en v6). Celle-ci intègre une interface de modélisation de processus (le Studio), un moteur d'exécution (bundle Tomcat sous debian) et une interface utilisateur (User XP). Bonita respecte la norme BPMN 2.0, ce qui permet notamment d'exporter/importer des processus issus d'autres applications.

2.2 Processus

Après étude du processus de validation des ordres de mission, nous l'avons modélisé dans le Studio de Bonita comme présenté ci-après.

2.2.1 Déroulement

Dans notre Environnement Numérique de Travail, l'agent dispose d'un lien pour saisir les informations concernant sa mission. Cette dernière est ensuite complétée par un gestionnaire financier qui rajoute les détails budgétaires. Le responsable de l'agent lui accorde l'autorisation d'absence, puis le responsable du budget valide l'ordre de mission. En cas de départ à l'étranger, et plus particulièrement dans les pays à risque, le président peut avoir à donner son autorisation. Pour les services centraux une validation N+1 (chef de service) a été ajoutée.

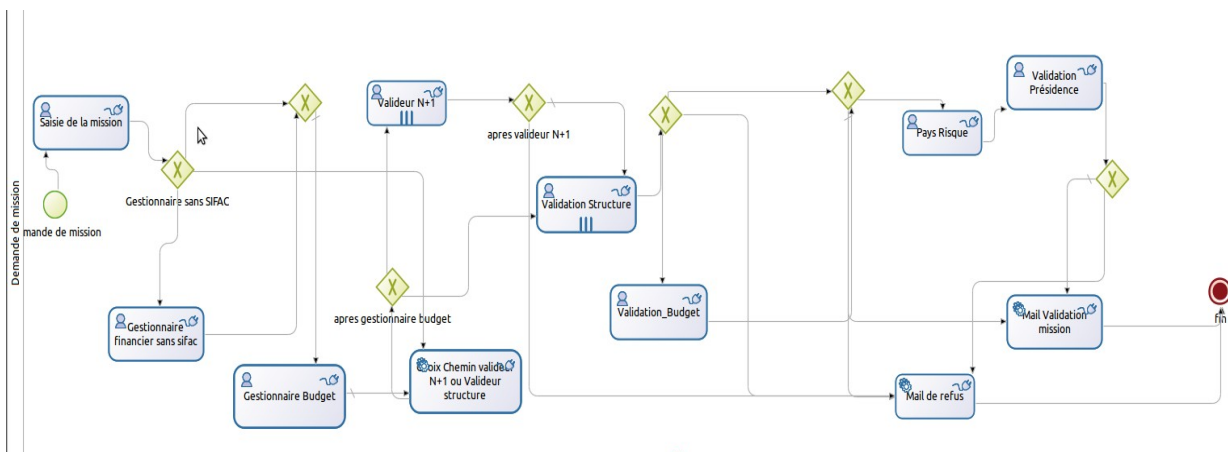


Figure 1: Processus modélisé validation OM

2.2.2 Exemple

Un enseignant de l'UFR Sciences Humaines (SH) part en mission pour son laboratoire LABOX.

- 1- L'enseignant saisie sa mission
- 2- Le gestionnaire comptable du laboratoire LABOX saisit les informations budgétaires (l'enseignant dispose de champs commentaires pour éventuellement donner son avis sur le financement)
- 3- Le directeur de l'UFR SH valide l'autorisation d'absence.

4- Le directeur du LABOX valide l'engagement financier.



Demande de mission

Saisie de la mission

Début: 7 nov. 2013 10:41 Fin: Priorité: **Normale**

Les demandes doivent être saisies une semaine avant le départ pour les missions en France métropolitaine et UE, et deux semaines avant le départ pour les missions hors UE et Dom Tom

Nom Panel

Prénom Mathieu

Email Mathieu.Panel@upmf-grenoble.fr

Date départ

Heure de départ de la résidence personnelle ou administrative 00:00

Date retour

Heure de retour de la résidence personnelle ou administrative 00:00

Autre Structure UPMF Autre structure UPMF

Motif du déplacement

Localisation France (hors Outre-Mer) UE Hors UE et Outre-Mer

Lieu de la mission

Informations Transport

Moyen(s) de transport Avion Train 2eme classe Train 1ere classe Véhicule personnel Véhicule administratif Autre

Commentaire sur le transport

Informations Mission

Mission financée par DSI-UPMF

Autre Structure UPMF Autre structure UPMF

Divers

Commentaire (éléments sur le budget, avance...)

Soumettre ma demande de mission

Figure 2: Formulaire demande de mission

2.2.3 Taches multi instanciées

Pour répondre au cas des agents travaillant à mi-temps dans plusieurs services et/ou UFR, les taches « validation N+1 » et « validation Structure » (Autorisation d'absence) sont multiinstanciées. Tous les responsables de l'agent doivent donner leur accord pour que le processus continue.

2.3 Référentiel



Gestionnaire financier Ordre de mission

Le responsable principal est en gras.

Structure	Type		Responsable (principal en gras)	
Fac. de Sciences Eco	Composante	+	JOSETTE (actif)	✗
			CHANTAL (actif)	✗
			NATALIA (actif)	✗
			CHRISTIANE (actif)	✗
			CHRISTIANE (actif)	✗
Fac. de Sciences Eco - Valence	Composante	+	JOSETTE (actif)	✗
			CHANTAL (actif)	✗
IAE Grenoble	Composante	+	HÉLÈNE (actif)	✗
			CATHERINE (actif)	✗
IAE Grenoble-Valence	Composante	+	CATHERINE (actif)	✗
			HÉLÈNE (actif)	✗
IEP Grenoble	Composante	+		
IUT Valence	Composante	+	MICKAEL (actif)	✗
			CECILE (actif)	✗
IUT2 Grenoble	Composante	+	AUDREY (actif)	✗
			MARIE-JOSÉ (actif)	✗

Figure 3: Les acteurs de la tâche Gestion financière par structure

Un enjeu important de ce projet est la configuration de l'application « ordre de mission », et surtout la sélection des acteurs de chaque tâche. Un outil de référentiel a donc été développé autour du projet des missions. Il permet de définir les acteurs des tâches du processus, mais également leur suppléant en cas d'absence et la gestion des cas particuliers (Directeur d'UFR validé directement par la présidence). L'outil référentiel est développé en PHP avec le framework cakePHP. Il évolue encore aujourd'hui pour la gestion de l'alimentation des personnels. Ces informations sont nécessaires dans tout processus, et étaient absentes de notre SI.

Sur la figure 3, l'IAE Grenoble fonctionne avec deux gestionnaires financiers en parallèles (les deux reçoivent toutes les demandes). L'IUT2 Grenoble utilise le mode suppléant, pour déléguer sa tâche la personne utilise l'interface suivante :

Déléguer provisoirement des droits dans les processus automatisés

Vous êtes responsable dans ces processus automatisés

Responsabilité	Sur	Délégation	Suppléants	Action
Valideur budget Ordre de mission	DSI-UPMF	non délégué		Déléguer !
Valideur structure Ordre de mission	TICE-EAD	non délégué		Déléguer !

Figure 4: Délégation

Ci-dessous la gestion des cas particuliers :

CHRISTIAN [REDACTED]	IAE Grenoble	Valideur structure Ordre de mission OM_VALID_STRUCT	SÉBASTIEN [REDACTED]	✘
CHRISTIAN [REDACTED]	IAE Grenoble-Valence	Valideur structure Ordre de mission OM_VALID_STRUCT	SÉBASTIEN [REDACTED]	✘

Figure 5: Cas particuliers

3 Intégration

3.1 Synchronisation annuaire LDAP

Pour le fonctionnement du processus, les utilisateurs du SI ont besoin d'être importés au sein de Bonita. La version opensource ne dispose pas de la synchronisation LDAP. Elle met cependant à disposition une API pour dialoguer avec le moteur. Nous avons donc développé le lien LDAP en JAVA. Ci-dessous un exemple de mise à jour de données d'un utilisateur via l'API identity.

```
StandardAPIAccessorImpl accessor1 = new StandardAPIAccessorImpl();

IdentityAPI identityAPI1 = accessor1.getIdentityAPI();

System.out.println(username+" Pas créé, probablement parcequ'il
exsiste on essaie de le mettre À jour"+e.toString()+"\n");

System.out.println("On fait une recherche sur le username :
"+username+"\n");

User u2 = identityAPI1.findUserByUsername(username);

System.out.println("USER : "+u2.toString());

identityAPI.updateUserByUUID(u2.getUUID(),username, prenom, nom, "",
"", "", null);

identityAPI.updateUserPersonalContactInfo(u2.getUUID(), mail, "",
"", "", "", "", "", "", "", "", "", "");
```

L'API nous apporte de nombreuses fonctionnalités intéressantes, comme récupérer l'état d'une tâche ou d'un processus en direct.

3.2 Connecteurs

Chaque tâche du processus a besoin d'informations : ses acteurs, mail pour notification, interrogation de base de données pour définir une condition du processus. Pour ce besoin Bonita met à disposition des connecteurs. Certains sont prédéfinis, par exemple Mysql ou LDAP, d'autres peuvent être complètement développés en JAVA ou Groovy Script. Ces connecteurs permettent un dialogue efficace avec les différentes briques du SI, ou bien avec des applications extérieures qui mettent à disposition des webservices.

3.2.1 Définition des acteurs d'une tâche

La définition des acteurs d'une tâche se fait dans notre processus par une requête SQL, sur la base référentiel. Ainsi tout changement d'acteur est piloté par le référentiel, et non dans le processus lui même.

3.2.2 Récupération des mails d'acteurs d'une tâche via Groovy script

Dans un connecteur Groovy Script, on peut récupérer des informations sur les acteurs d'une tâche. Ci-dessous on alimente la variable « to » avec les adresses mails de tous acteurs de la tâche courante:

```
import org.ow2.bonita.util.AccessorUtil

def candidates =
AccessorUtil.getQueryRuntimeAPI().getTaskCandidates(activityInstance
.getUUID())
def to="" ;
for(user in candidates){
if(to!="") to+=", "
to+=AccessorUtil.getIdentityAPI().getUser(user).getEmail()
}
return to
```

Ceci est utilisé pour la notifier une personne de la tâche qu'elle a à faire.

3.3 Evolution / Mutualisation

La construction du référentiel est un enjeu fort du projet, son alimentation et sa mise à jour sont essentielles. Il nous permettra de construire d'autres processus ou d'en alimenter des existants (validation des congés). Le couple moteur de workflow/référentiel permet la mise en œuvre efficace de la simplification des procédures. La mutualisation des développements entre projets est intéressante, tous les connecteurs développés enrichissent une bibliothèque et sont largement réutilisables. Le moteur est très communiquant et autorise le développement d'outils de suivi des processus.