

## Tâches 3.2 et 3.3 :

### Editeurs de capacités et d'objectifs : suite collaborative OpenPaaS

#### A propos:

La suite livrée constitue les livrables 3.2.4 et 3.3.4 dont le but est de fournir des éditeurs de collaborations (trois modules web en tout) dans le cadre du projet de recherche OpenPaaS :

- un éditeur de profil (ou de capacités) **profil\_modeler**, visant à aider un utilisateur à définir les activités métiers qu'il est capable de mettre à disposition sur la plateforme OpenPaaS ;
- un éditeur d'objectifs **objective\_modeler**, qui permet à un utilisateur d'explicitier les objectifs de collaboration donc il a besoin et auxquels la plateforme se propose de répondre en déduisant un processus collaboratifs faisant appel à l'ensemble des activités métiers disponibles sur la plateforme.
- Un éditeur de processus collaboratifs **collaborative\_process** qui permet de visualiser le processus qui a été déduit par l'éditeur précédent, et de l'éditer au besoin.

Ces éditeurs permettent à l'utilisateur d'utiliser des outils uniquement graphiques (permettant une expérience plus intuitive) visant à exposer des connaissances sur les contextes de collaboration (capacités métiers ou objectifs de collaboration). Les modèles produits sont des fichiers « .gem », contenant un code XML lisible et éditable par des éditeurs type NotePad.

#### Exécution :

##### *Ubuntu*

Aller dans le dossier /bin et lancer le script startup.sh

##### *Windows*

Aller dans le dossier /bin et lancer le script startup.bat

##### *Mac*

Aller dans le dossier /bin et , dans le terminal lancer la commande :

```
java -ea -Xmx1024m -jar './lib/gind-openpaas-suite-1.0-SNAPSHOT.jar'
```

Si erreur sous Mac : essayer en installant la JVM 1.8.

Pour éteindre Ctrl+C.

#### Accéder aux éditeurs:

Automatiquement au lancement, une page doit s'ouvrir permettant d'avoir un accès soit à l'éditeur de capacités, soit à l'éditeur d'objectifs.

Ce qui nous intéresse dans le cadre de l'intégration des éditeurs concerne plutôt l'accès à l'un ou à l'autre sur demande, grâce à une url spécifique. Cette url est la suivante :

- Pour l'éditeur de capacités :
  - [http://192.168.33.97:8870/webjars/profil\\_modeler/profil\\_modeler.html?token=3MT1tgMrGKO46kml9mHOKL/Haj+W3spRBBCVgnrfFwmZFbMH2Jsuj4VlcZxHhDt84/EsrXfs3+xlId6hht81fQ](http://192.168.33.97:8870/webjars/profil_modeler/profil_modeler.html?token=3MT1tgMrGKO46kml9mHOKL/Haj+W3spRBBCVgnrfFwmZFbMH2Jsuj4VlcZxHhDt84/EsrXfs3+xlId6hht81fQ)
- Pour l'éditeur d'objectifs :
  - [http://192.168.33.97:8860/webjars/objective\\_modeler/objective\\_modeler.html?token=3MT1tgMrGKO46kml9mHOKL/Haj+W3spRBBCVgnrfFwmZFbMH2Jsuj4VlcZxHhDt84/EsrXfs3+xlId6hht81fQ](http://192.168.33.97:8860/webjars/objective_modeler/objective_modeler.html?token=3MT1tgMrGKO46kml9mHOKL/Haj+W3spRBBCVgnrfFwmZFbMH2Jsuj4VlcZxHhDt84/EsrXfs3+xlId6hht81fQ)
- Pour l'éditeur de processus collaboratifs :
  - [http://192.168.33.97:9030/webjars/ioda/collaborative\\_process/collaborative\\_process.html?token=3MT1tgMrGKO46kml9mHOKL/Haj+W3spRBBCVgnrfFwmZFbMH2Jsuj4VlcZxHhDt84/EsrXfs3+xlId6hht81fQ](http://192.168.33.97:9030/webjars/ioda/collaborative_process/collaborative_process.html?token=3MT1tgMrGKO46kml9mHOKL/Haj+W3spRBBCVgnrfFwmZFbMH2Jsuj4VlcZxHhDt84/EsrXfs3+xlId6hht81fQ)

L'url contient donc l'url de base d'accès à l'appli web avec un token spécifique (qui ne change jamais) passé en paramètre GET. Ce token correspond à l'id de l'utilisateur qui est dans notre cas enregistré par défaut (openpaas admin) pour faciliter l'intégration sur la plateforme.

## Intégration au sein du RSE :

L'intégration au sein du RSE intervient de deux façons différentes :

- **Lors de la lecture d'un fichier .gem**
- **Lors de l'écriture d'un fichier .gem**

### Lecture d'un fichier .gem

Afin de pouvoir lire un fichier .gem sur un éditeur, on peut soit l'importer à partir du poste client (ce qui n'est pas le plus utile dans le cas de l'intégration des éditeurs en tant que SaaS sur la plateforme), soit l'appeler via une requête Rest HTTP GET.

*Exemple d'appel avec la méthode GET :*

[http://172.20.10.223:8870/webjars/profil\\_modeler/profil\\_modeler.html?token=3MT1tgMrGKO46kml9mHOKL/Haj+W3spRBBCVgnrfFwmZFbMH2Jsuj4VlcZxHhDt84/EsrXfs3+xlId6hht81fQ&url=http://openpaas:9002/resources/example7.gem](http://172.20.10.223:8870/webjars/profil_modeler/profil_modeler.html?token=3MT1tgMrGKO46kml9mHOKL/Haj+W3spRBBCVgnrfFwmZFbMH2Jsuj4VlcZxHhDt84/EsrXfs3+xlId6hht81fQ&url=http://openpaas:9002/resources/example7.gem)

*La partie verte concerne l'appel de l'appli web.*

*La partie rouge concerne le passage en GET du Token de sécurité permettant d'accéder au modeler (Token fixe dans notre cas).*

*La partie bleue concerne l'appel en GET du fichier .gem à lire.*

**N.B.** Le fichier .gem à lire doit être placé sur un serveur http accessible. Soit l'accès est non sécurisé comme l'exemple ci-dessous et l'appel est de la forme « **file:///path/file.gem** ». Soit Linagora souhaite utiliser, par exemple, un **service REST** (permettant de sécuriser l'appel aux modèles stockés sur le serveur http) qui est appelé directement dans le

paramètre GET « url », et qui retourne le fichier à lire. Auquel cas, on pourrait avoir une url du type : <http://openpaas:9003/services/getResources?opToken=04&rsclId=45>

### Cas de la création d'un nouveau modèle

Dans plusieurs cas, l'utilisateur est amené à créer un nouveau modèle :

- **Création d'un nouveau profil** : l'utilisateur doit décrire ses capacités sur un nouveau modèle. L'idée est, dans le RSE développé par Linagora, d'ajouter un bouton qui
  - (i)Crée un nouveau fichier .gem contenant une enveloppe de modèle vide (en pièce jointe de ce readme)
  - (ii)Génère un instanceId aléatoire/unique qui permet de « reconnaître le modèle » : une balise est prévue et doit être remplie à cette effet dans chaque fichier .gem (voir enveloppe d'exemple jointe)
  - (iii)Génère les propriétés globales suivantes :

```
<ns22:globalProperties>
  <ns22:property name="organization_id" value="organization_id"/>
  <ns22:property name="organization" value="organization_name"/>
  <ns22:property name="community_id" value="community_id"/>
  <ns22:property name="community" value="community_name"/>
  <ns22:property name="user_id" value="user_id"/>
  <ns22:property name="user" value="user_name"/>
</ns22:globalProperties>
```
  - (iv)Stocke en base de donnée l'instanceId d'un modèle et l'url correspondante sur le serveur http où sont stockés ces modèles : cela permet ensuite lors de l'édition d'un modèle de sauvegarder le fichier au bon endroit. (voir section : [Ecriture d'un fichier .gem](#))
  - (v)Finalement appelle l'appli web de l'éditeur pour lire le fichier, et permettre à l'utilisateur de l'éditer.
- **Création d'une nouvelle opportunité** : même principe que la création du nouveau profil, en utilisant l'appli web objective\_modeler au lieu de profil\_modeler

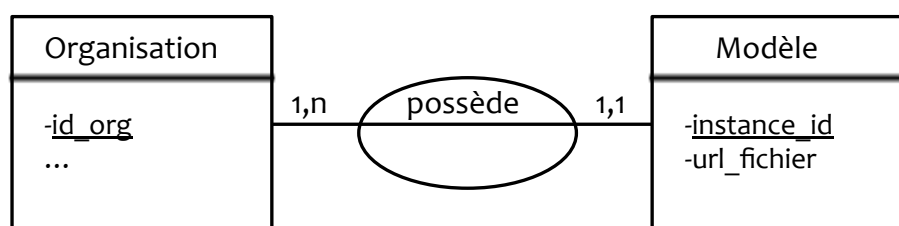


Figure 1 : MCD minimaliste.

### Ecriture d'un fichier .gem

Afin de mettre en place un service d'écriture permettant la mise à jour des modèles, nous proposons de mettre en place dans les éditeurs un bouton qui envoie le fichier .gem à un web service de Linagora (soit un service Rest avec une opération en HTTP Post pour pouvoir pousser le modèle sur le serveur, soit un service soap avec une opération de stockage de modèle adéquate). Le fichier .gem contiendra toujours

l'instance-id, qui permet et suffit du côté RSE pour retrouver l'url d'emplacement du fichier sur le serveur http où sont stockés les modèles (cf. MCD minimaliste ci-dessus).

## Utilisation des éditeurs de capacités et d'objectifs pour OpenPaaS

### Création de profils : description des capacités de chaque communauté d'OpenPaaS

#### Edition du profil

1. Dans la palette, sélectionner « capability », ou directement sur la partie canvas (jaune clair).
2. Double cliquer sur la « capability » affichée dans le canvas pour exprimer ses propriétés :
  - a. Nom
  - b. Propriétés d'affichage (ex. couleur de la boîte)
  - c. Lien à l'ontologie de collaboration
  - d. Propriété non-fonctionnelles (ex. prix, délai de production, coûts...)
3. Dans la palette ou sur le canvas, cliquer sur « Input » et « Output » pour exprimer les entrées et sorties de la « capability ».
  - a. Nom
  - b. Propriétés d'affichage.
  - c. Lien à l'ontologie domaine

#### Mise à jour de l'ontologie de collaboration

Les profils permettent de mettre à jour l'ontologie de collaboration, comme illustré sur la Figure 2, dans laquelle trois capacités sont proposées par des communautés.

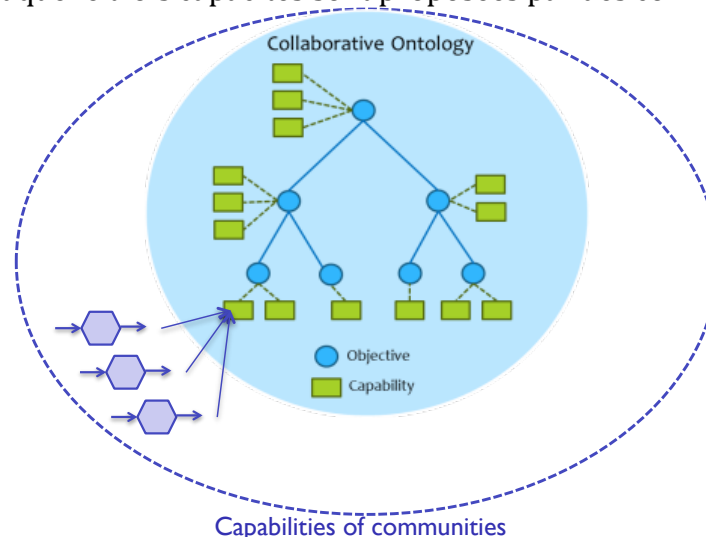


Figure 2: Intégration des capacités de communautés à l'ontologie de collaboration.

L'ontologie offre initialement une base de connaissances pérenne, structurée de la façon suivante : des objectifs de collaboration peuvent être remplis soit par un ensemble de capacités à exécuter, soit être décomposés eux-mêmes en sous-objectifs ou en capacités. L'éditeur de profils permet aux communautés d'OpenPaaS d'exprimer les capacités qu'elles désirent proposer pour des collaborations futures : elles proposent donc une instantiation des capacités pérennes de l'ontologie de collaboration (en en indiquant les

propriétés complètes : non-fonctionnelles ainsi qu'entrées requises et sorties i.e. résultats des capacités).

*Dans le menu de l'éditeur : cliquer sur « Populate ontology ».*

### **Renvoi des compétences au RSE**

Outre le fait de mettre à jour l'ontologie de collaboration, l'éditeur de profil permet de décrire les « skills » ou compétences d'une communauté, qui doivent être affichés sur le RSE. Ainsi un déploiement des modèles permet d'envoyer les données de l'éditeur au RSE.

*Dans le menu de l'éditeur : cliquer sur File > « Deploy ».*

## **Création d'opportunités : description d'opportunités de collaboration et déduction de processus correspondants**

### **Edition des opportunités**

1. *Dans la palette, sélectionner «objective», ou directement sur la partie canvas (jaune clair).*
  - a. *Nom*
  - b. *Propriétés d'affichage (ex. couleur de a boîte)*
  - c. *Lien à l'ontologie de collaboration*
  - d. *Objectifs non-fonctionnels*
2. *Dans la palette ou sur le canvas, cliquer sur « domain » pour exprimer les entrées et sorties du «domain».*
  - a. *Nom*
  - b. *Propriétés d'affichage.*
  - c. *Lien à l'ontologie domaine*

### **Déduction de processus inter-communautés**

A partir du lien donné entre les opportunités proposées sur l'éditeur et l'ontologie de collaboration, le système est capable de connaître quels sont les objectifs de l'ontologie de collaboration pérenne à remplir. On peut donc trouver les ensembles de capacités permettant de répondre aux opportunités, et grâce aux éditeurs de profils, quelles sont les communautés capables d'exécuter celles-ci. Une optimisation lors de la déduction de processus permet de renvoyer à l'utilisateur ayant proposé l'opportunité, les processus répondant au mieux. Un processus (au format BPMN) correspond aux données suivantes :

- Les différents acteurs
- Les activités exécutées de chaque acteur
- Le séquençement des activités

Le meilleur processus est retenu et la liste des acteurs et leurs performances non-fonctionnelles sont renvoyées sous forme d'un document json au RSE, qui peut ainsi afficher les résultats obtenus.

*Dans le menu de l'éditeur : cliquer sur Action > « Deduce process ». Un fichier .pco est téléchargé sur le poste client.*

### **Lecture et édition du processus collaboratif**

L'éditeur de processus collaboratif permet tout simplement d'afficher le processus déduit à l'étape précédente, et de le visualiser et l'éditer si l'utilisateur souhaite faire des

modifications suite à cette déduction pour mieux l'adapter à ses méthodes propres de t  
*Dans le menu de l'éditeur : cliquer sur File > « Import ».*  
*Sélectionner le fichier .pco qui a été téléchargé sur le poste à l'étape précédente.*  
*L'ensemble des outils BPMN courants (tâches, pools, gateways) sont disponibles dans le menu Palette, pour toute modification du processus.*