

# TD UML 2013 - Emmanuel Pichon

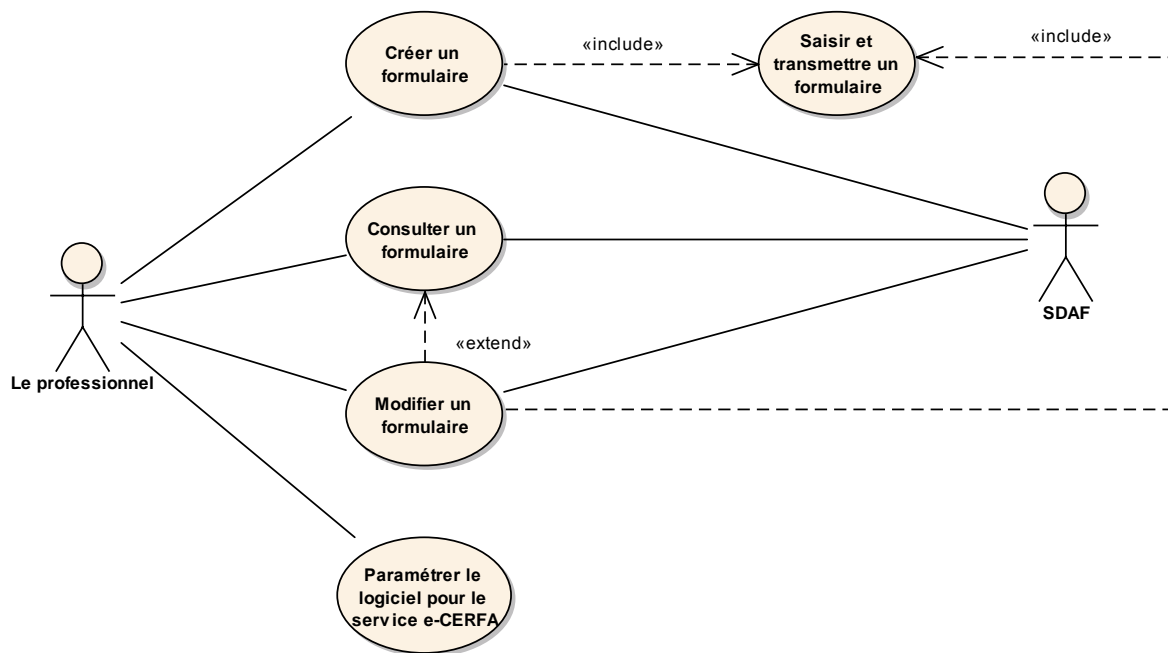
## Analyse d'impacts

### Eléments de corrigé

---

## 1 Impacts sur le recueil des besoins et l'analyse

### 1.1 Diagramme de cas d'utilisation

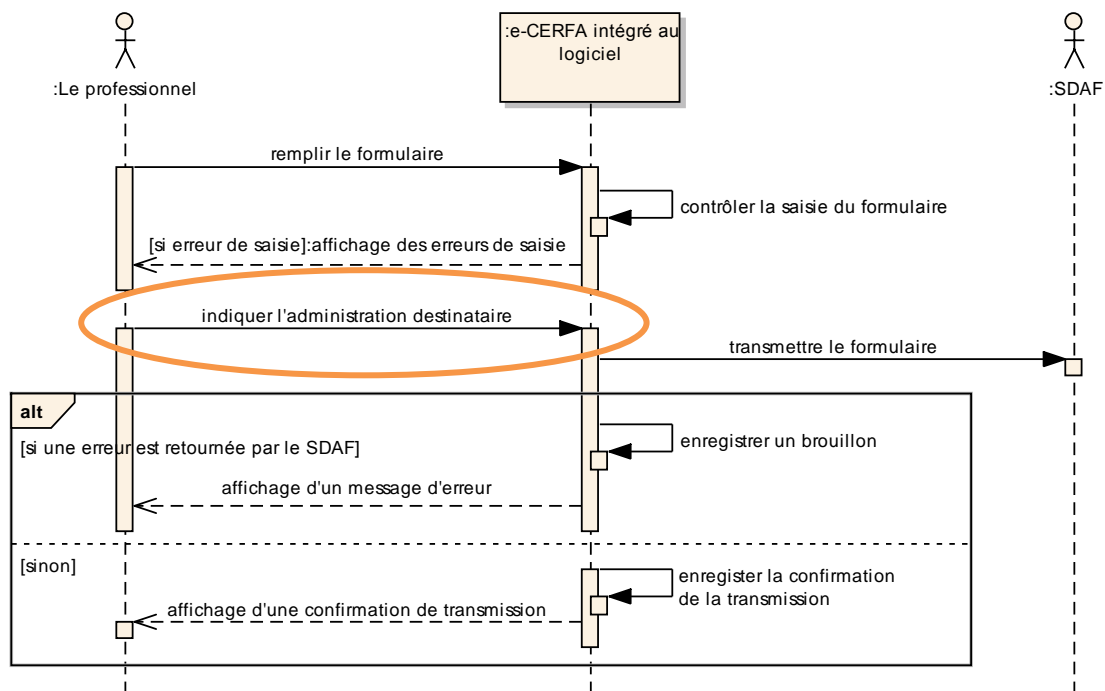


Pas d'impact sur ce diagramme.

A priori, le seul cas d'utilisation impacté (en termes de documentation) est « Saisir et transmettre un formulaire ».

Le diagramme de séquence correspondant à ce cas d'utilisation est à vérifier.

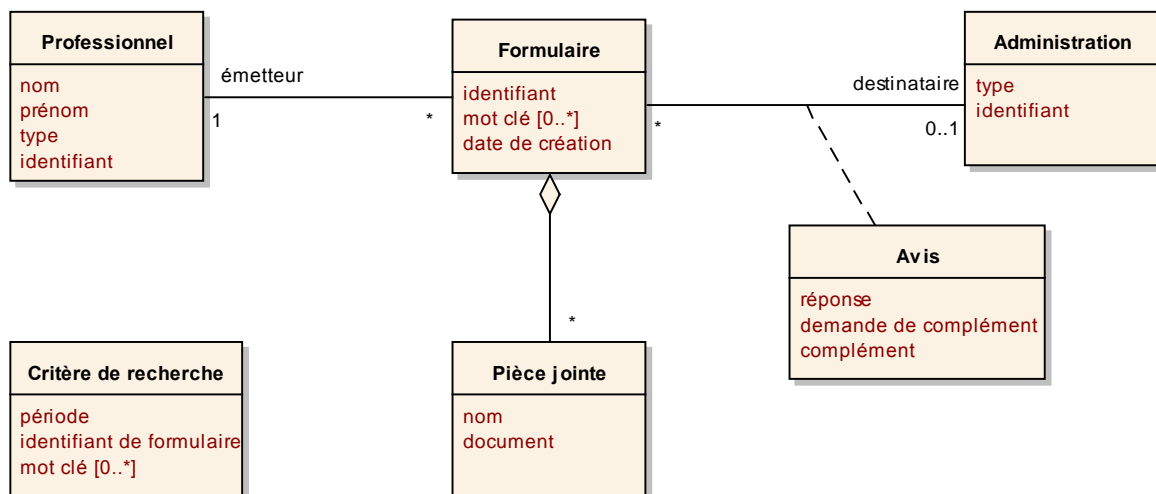
## 1.2 Diagramme de séquence « Saisir et transmettre un formulaire »



L'évolution impacte le message « indiquer l'administration destinataire » qui devient au choix :

- « indiquer l'administration ou le professionnel destinataire »,
- « indiquer le destinataire ».

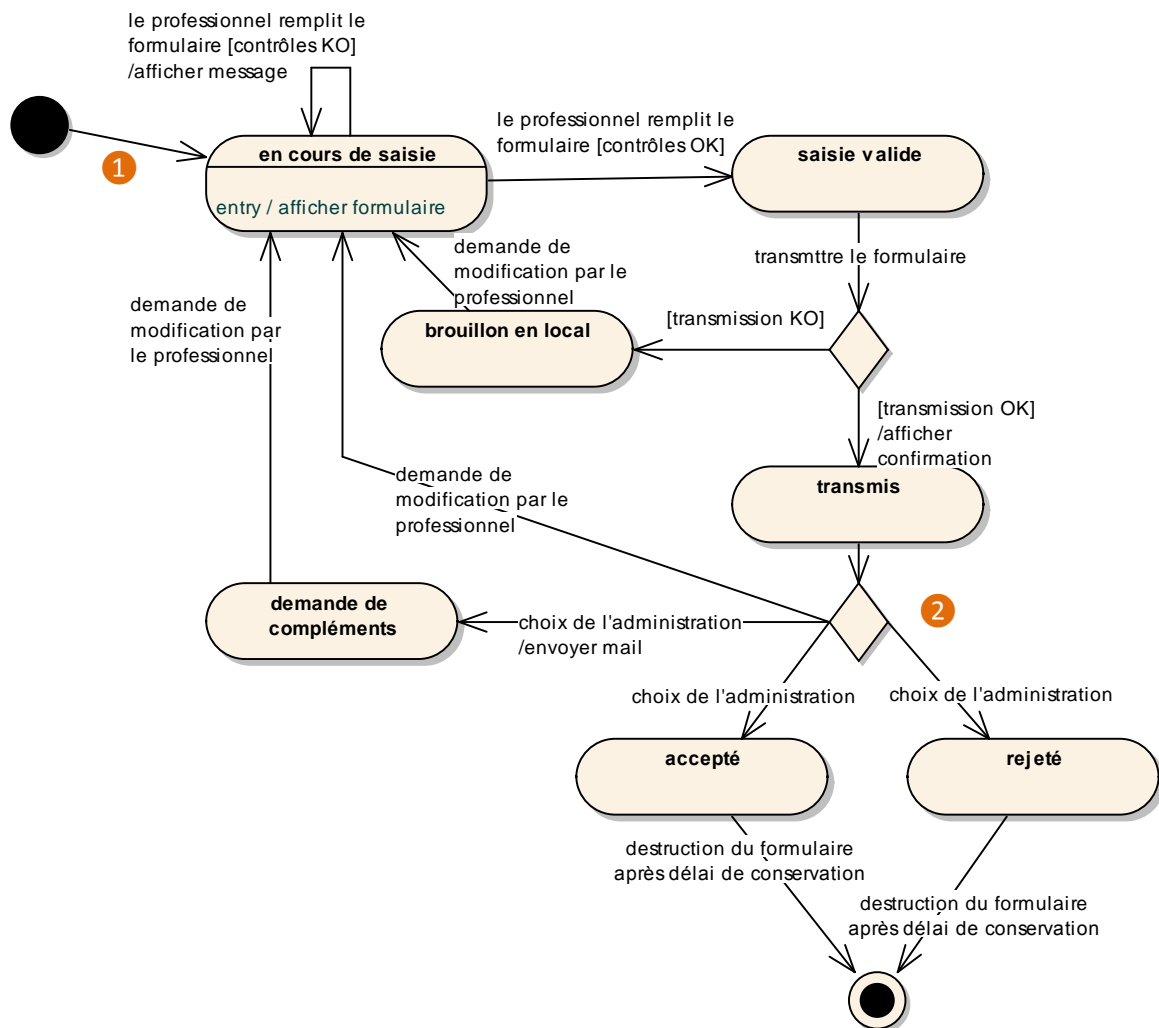
## 1.3 Diagramme de classes



L'évolution implique l'identification d'une association supplémentaire entre « Formulaire » et « Professionnel » avec un rôle « destinataire ».

Hypothèse simplificatrice pour le TD : lorsqu'un professionnel reçoit un formulaire, il ne peut pas donner son avis sur ce formulaire.

## 1.4 Diagramme des états d'un formulaire



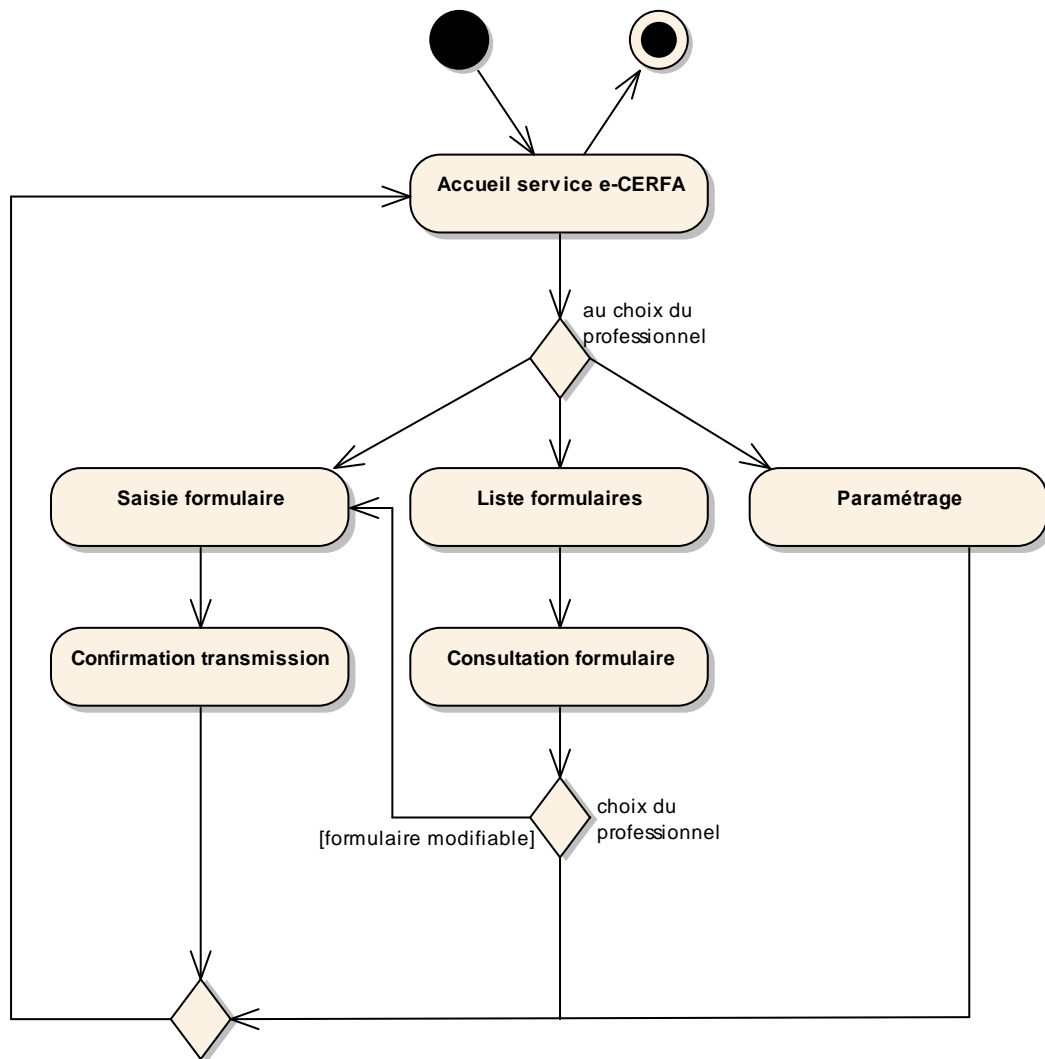
L'évolution permettant l'envoi d'un formulaire à un professionnel implique :

- l'ajout d'une fin suite à l'état transmis, avec une condition « destinataire = un professionnel » (cf. point 2 dans le diagramme),
- la modification des autres conditions pour ajouter « destinataire = une administration ».

NB : ce diagramme présente les états du formulaire dans le logiciel du professionnel émetteur de ce formulaire.

Si l'on souhaite intégrer la consultation d'un formulaire émis par un autre professionnel dans notre logiciel, un état « reçu » est à ajouter directement après le point de départ. Les conditions de passage aux autres états sont à étudier en fonction des possibilités que l'on souhaite offrir à l'utilisateur. Par exemple, s'il veut pouvoir consulter, modifier puis retransmettre le formulaire, il faut prévoir une transition entre l'état « reçu » et l'état « en cours de saisie » (cf. point 1 dans le diagramme).

## 1.5 Diagramme d'activités « enchainement des écrans »

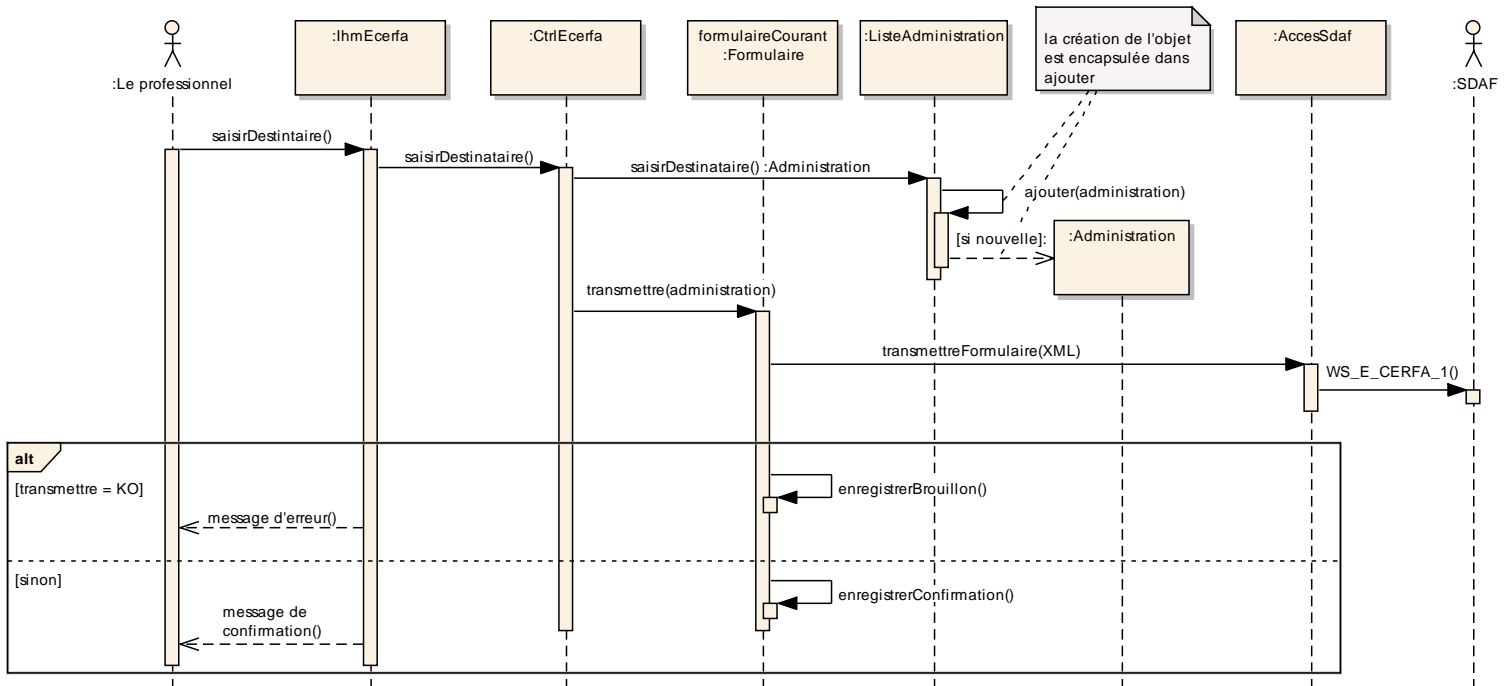


L'évolution n'a pas d'impact sur ce diagramme.

Ce diagramme nous permet d'identifier un impact concernant l'écran « Saisie formulaire » dans lequel le professionnel choisit le destinataire.

## 2 Impacts sur la conception

### 2.1 Diagramme de séquence « Transmettre un formulaire »

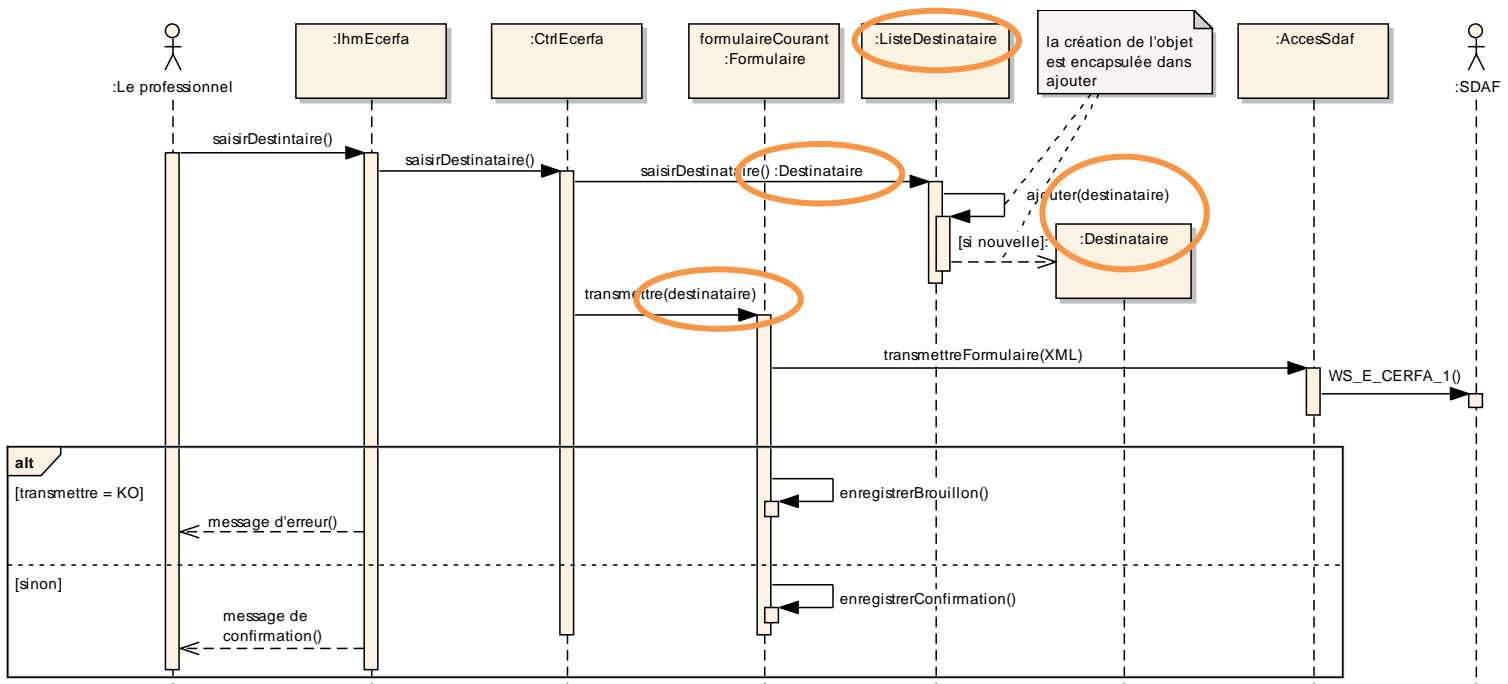


Deux choix de conception sont possibles pour intégrer l'évolution :

- Dupliquer le schéma si dessus et remplacer Administration par Professionnel et ListeAdministration par ListeProfessionnel,
- Généraliser Administration en Destinataire et ListeAdministration en ListeDestinataire.

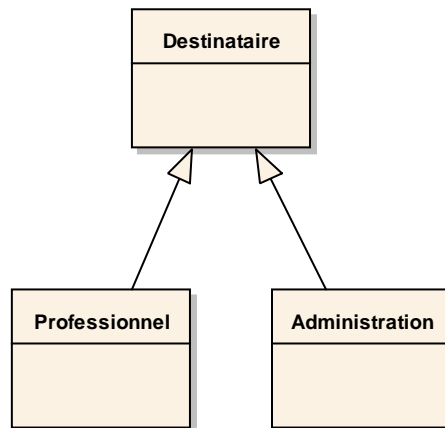
La deuxième solution est la plus « objet » car elle est plus encapsulée et elle tire partie de l'héritage.

Un autre avantage est de permettre l'ajout d'un autre type de destinataire en limitant les impacts.



## 2.2 Diagramme de classes

La généralisation se modélise comme indiqué ci-dessous.



NB : il faut penser à ventiler les opérations et les attributs définis initialement dans la classe Administration. On déplace les éléments communs dans la classe Destinaire (non illustré).

Question : peut-on placer la classe Professionnel initiale dans le diagramme ci-dessus ?

La classe Professionnel identifiait initialement l'utilisateur de notre logiciel. Par exemple, l'opération validerSaisieParametrage n'est valable que pour l'utilisateur du logiciel. Dans ce cas, on peut distinguer deux classes : Professionnel et Utilisateur. On peut, par exemple, définir Utilisateur comme une sous-classe de Professionnel. Ceci permet de mutualiser les attributs et de spécialiser l'opération validerSaisieParametrage en la déplaçant dans la classe Utilisateur. Le diagramme ci-dessous présente les classes modifiées et les classes liées à celles-ci.

