

<p><b><i>Projet 1 d'UML</i></b> <b><i>Agence de presse</i></b></p>
--

**Le Contexte**

L'entreprise concernée est l'agence de presse **AFP**<sup>1</sup>.

Le métier de AFP est de :

- a) suivre l'actualité mondiale sur le terrain
- b) rédiger des news (articles de presse, dossier de synthèses) sur les thèmes d'actualité,
- c) pour chaque news, envoyer ces news à ceux de ses clients intéressés par le thème des news.

**L'application à réaliser**

L'AFP souhaite informatiser le processus de production de ses news, afin de mieux planifier le travail de ses collaborateurs. Ces collaborateurs sont des journalistes, des traducteurs et des correcteurs.

L'objectif de l'application est

- (a) d'améliorer et optimiser la répartition du travail
- (b) de faciliter la gestion des collaborateurs et
- (c) d'améliorer la rapidité de rédaction et donc la réactivité de l'entreprise AFP.

**Acteurs et usages**

L'application est utilisée par :

- les responsables de l'agence (rédacteur en chef notamment)
- les collaborateurs
- Les administrateurs du système

**Les responsables :**

Ils décident de créer une news sur un thème donné, puis planifient et valident le travail.

Valider le travail signifie : accepter la rédaction de la news et permettre ainsi le démarrage des tâches de correction et de validation. Si le responsable n'est pas d'accord avec la rédaction, il envoie un mail au rédacteur.

Planifier le travail signifie :

- (a) créer les tâches à réaliser (rédiger, corriger et éventuellement traduire),
- (b) définir les dates de fin
- (c) affecter chaque tâche à un collaborateur compétent pour cette tâche.

Le point (c) implique qu'un collaborateur soit disponible (les collaborateurs ne peuvent pas se voir affecter trop de tâches).

---

<sup>1</sup> Toute ressemblance avec une entreprise réelle est bien évidemment le fruit du hasard.

---

Les responsables gèrent également les collaborateurs : ils définissent les compétences de chacun des collaborateurs, et les thèmes d'actualité sur lesquels les rédacteurs sont qualifiés.

### Les collaborateurs :

Les collaborateurs (rédacteurs, correcteurs, traducteurs) prennent connaissance des tâches qui leur sont affecté, puis signalent au système quand ils ont fini leur travail. Ils sont répartis dans le monde entier.

### Les administrateurs :

Ils gèrent les collaborateurs, et mettent en œuvre le système (initialisation, démarrage des batchs...).

### Le système

Les responsables sont alertés par le système sur des situations critiques (notamment : lorsqu'une news n'est pas encore rédigée alors qu'il reste moins de x heures avant la date de livraison).

L'envoi d'une news au(x) client(s) s'effectue par messagerie électronique. L'envoi des news n'est effectué que si le client est en règle vis à vis de son contrat et de ses factures.

Les news déjà diffusées sont archivées un an après leur diffusion, et supprimées du système (c'est à dire supprimées dans la base de données relationnelles).

### Le périmètre technique de l'application

L'application est reliée au système de gestion de contrat et de facturation, disponible sur un mainframe.

L'application s'appuie sur les investissements déjà réalisés par l'entreprise : OS WINDOWS, serveur de base de données ORACLE.

Les collaborateurs sont soit des salariés de l'entreprise, soit des indépendants (« pigistes »).

Du fait de la répartition géographique de ces collaborateurs, il est nécessaire de s'appuyer sur une architecture **WEB**. L'application est donc un **Extranet**.

¶ Une évolution ultérieure pourra être d'ouvrir le système pour qu'un client puisse

- (a) créer lui-même une nouvelle news
- (b) suivre l'avancement de cette news.

### La méthodologie

Compte tenu de l'orientation WEB indispensable, il est décidé d'appliquer les standards suivants :

- méthode UP<sup>2</sup> (Unified Process)
- langage de modélisation UML
- langage de programmation JAVA
- architecture J2EE

infrastructure logicielle WINDOWS, ORAC

---

<sup>2</sup> Un processus unifié est un processus de développement logiciel construit sur UML, il est itératif, centré sur l'architecture, conduit par les cas d'utilisation et piloté par les risques. La gestion d'un tel processus est organisée suivant les 4 phases suivantes : pré étude, élaboration, construction et transition.

**Etapes à suivre**

- 1- Effectuer l'analyse du contexte, et décomposer le système en modules et packages ;
- 2- Pour chaque package, élaborer un diagramme de cas d'utilisation (UC) ;
- 3- Pour chaque UC, proposer des diagrammes d'objets. Sélectionner seulement les objets mis en jeu dans le UC ;
- 4- Pour chaque package, élaborer le diagramme des classes (statique) ;
- 5- Pour chaque UC, élaborer un diagramme de séquences ;
- 6- Intégrer les méthodes tirées des diagrammes de séquences dans les diagrammes des classes (dynamiques) ;
- 7- Pour les classes, dont le cycle de vie de leurs instances revêt une importance dans l'application, élaborer un diagramme d'états / transitions ;
- 8- Pour les méthodes, dont la mise en œuvre est relativement complexe, élaborer un diagramme d'activités ;
- 9- Elaborer le diagramme de composants pour l'application ;
- 10- Elaborer le diagramme de déploiement pour l'application ;
- 11- Générer les classes (.java) de l'application ;
- 12- Générer le modèle physique de données.

**NB :** Utiliser Entreprise Architect pour les différents modèles (sauf pour le diagramme de contexte, et les diagrammes d'objets) .