

Le livre débute toujours par la section 1, mais une ou plusieurs sections peuvent mener à la fin du livre, soit parce que le personnage est mort, soit par une fin plus heureuse à l'aventure.

On propose de réaliser un logiciel d'aide à la création de livres dont vous êtes le héros.

L'auteur crée un livre (en fixant son titre...). Il peut alors construire une section en rédigeant son texte. Il peut aussi définir un nouvel enchaînement entre sections. Les enchaînements peuvent être annotés par une condition, qui définit les prérequis (objets obtenus) pour franchir cet enchaînement. L'ensemble est présenté à l'auteur dans une interface graphique qui permet de visualiser le graphe complet des enchaînements possibles. Une analyse du graphe par l'outil permet de détecter les sections inatteignables, parce qu'il n'existe pas de chemin depuis la section 1 à la section incriminée.

Une fois le graphe construit, l'auteur peut demander la génération d'une version imprimable du livre. Pour ce faire l'outil numérote les sections de façon aléatoire (sauf la section 1) puis génère un texte qui capture les enchaînements. Il peut également demander la génération d'un jeu de pages HTML permettant à un lecteur de jouer le livre interactivement en ligne. Chaque page HTML correspond à une section, et seuls les choix actifs (dont la condition est remplie) sont visibles.

L'auteur peut ensuite facilement déployer ces pages web sur un quelconque hébergeur web pour les offrir au public. Il suffit enfin au joueur de visiter avec un navigateur web la section 1 du livre pour entamer l'aventure ; la gestion de l'inventaire est réalisée par des cookies HTTP de durée la session.

Question 2.1 : (3 pts) Réalisez le diagramme de cas d'utilisation du système.

Attention à la cohérence avec la Question 2.2 ci-dessous, qui exige la modélisation d'un cas d'utilisation « Définir enchaînement ».

Commentez et/ou annotez un minimum le diagramme.

Question 2.2 : (3 pts) Réaliser la fiche détaillée du ou des cas d'utilisation(s) couvrant les étapes qui permettent à un auteur de créer un enchaînement entre sections du LDVH. Par cohérence, on demande de modéliser ce cas d'utilisation en question 2.1.

Question 2.3 : (4 pts) Proposez un diagramme de classes d'analyse pour ce problème.

On ne détaillera pas les opérations de la classe (fictive) représentant le système.

3. Conception : SMTP a simple mail transfer protocol [7 Pts]

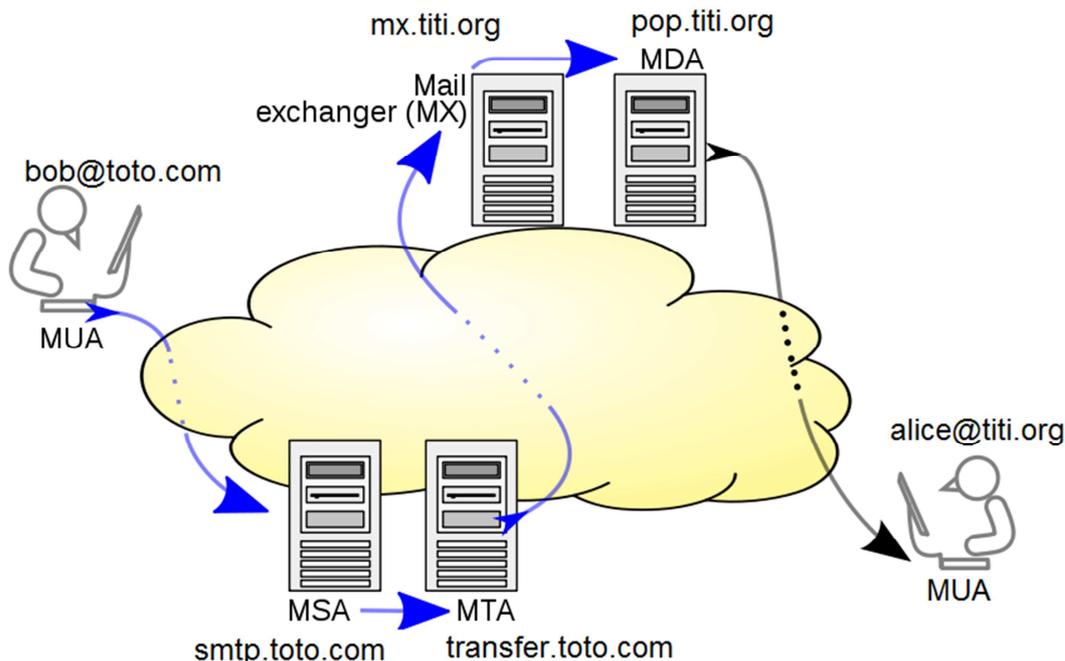


Figure 2 : Fonctionnement du mail

Simple Mail Transfer Protocol (SMTP, littéralement « protocole simple de transfert de courrier ») est un protocole de communication utilisé pour transférer le courrier électronique (courriel ou email) vers les serveurs de messagerie électronique.

SMTP est un protocole assez simple (comme son nom l'indique). On commence par spécifier l'expéditeur du message, puis le ou les destinataires d'un message, puis, en général après avoir vérifié leur existence, le corps du message est transféré.

SMTP ne permet pas de récupérer à distance des courriels arrivés dans une boîte aux lettres sur un serveur. Les standards Post Office Protocol (POP) et IMAP ont été créés dans ce but.

L'email est soumis par un client mail (MUA, mail user agent) au serveur mail de son domaine (typiquement smtp.domaine.net, appelé MSA, mail submission agent) en utilisant SMTP. A partir de là, le MSA livre le mail à l'un de ses agents de transfert (MTA, mail transfer agent) en utilisant SMTP. Les MTA peuvent jouer divers rôles, comme le filtrage de SPAM ou de contenus dangereux. L'utilisation d'une chaîne de MTA permet de distribuer la charge sur plusieurs serveurs au besoin. Chaque MTA ajoute à l'en-tête du message son nom, la date et les actions faites sur le message, ce qui explique la longueur et le contenu des en-têtes internet de vos mails. Chaque MTA est à la fois serveur et client pour le protocole SMTP.

La figure 2 illustre le fonctionnement de l'acheminement du mail. Toutes les flèches sauf celle liant le MDA au destinataire final sont implémentées par SMTP. La dernière étape est communément gérée via le protocole POP ou IMAP. Il peut y avoir plusieurs MTA qui forment une chaîne, dans le domaine émetteur (entre le MSA et le MTA frontière) comme dans le domaine récepteur (interposés entre le MX et le MDA).

Pour chaque destinataire du message, le MTA situé à la frontière du domaine source doit localiser le serveur de mail d'échange (MX, Mail eXchanger) du domaine cible. Pour cela, il s'appuie sur le service DNS pour résoudre l'adresse du MX, i.e. défini par la partie à droite de « @ » dans l'adresse email. Le MTA frontière se connecte alors comme un client SMTP au MX du domaine cible.

Si le MX accepte le message, il le transfère (potentiellement à travers une nouvelle chaîne de MTA qui réalise des filtres) à un agent qui va stocker les mails (MDA, mail delivery agent), et qui offre un service comme POP aux utilisateurs finaux pour réceptionner leurs mails.

Le protocole SMTP est un protocole textuel simple, comme le protocole POP étudié à l'examen de Janvier. Nous faisons ici abstraction de cette couche textuelle, qui pourrait être gérée à l'aide des DP Proxy Distant et Adapter comme on l'a fait pour POP afin d'avoir une modélisation orientée objet.

Question 3.1 (2 points)

On donne l'interface ISMTP qui permet de capturer le fonctionnement d'un serveur SMTP. On rappelle également les grandes lignes de l'interface IPOP : « list » permet d'obtenir une liste des mails stockés, et « retr » permet de lire un mail donné.



Sur un diagramme de composants, représentez les composants correspondant au MUA, MSA, MTA et MDA.

Question 3.2 (2 points)

Sur un diagramme de structure interne, représentez la configuration de la figure 2 de l'introduction.

Question 3.3 (3 points)

Montrez par un diagramme de séquence mettant en jeu les composants instanciés dans la question 3.2 les échanges nécessaires pour acheminer un email « coucou » du MUA émetteur (bob@toto.com) au MUA de destination (alice@titi.org).