Fiche de révisions - MOD et autorisations

Rédigé par : Jimmy Paquereau

1. Le Modèle Organisationnel des Données (MOD)

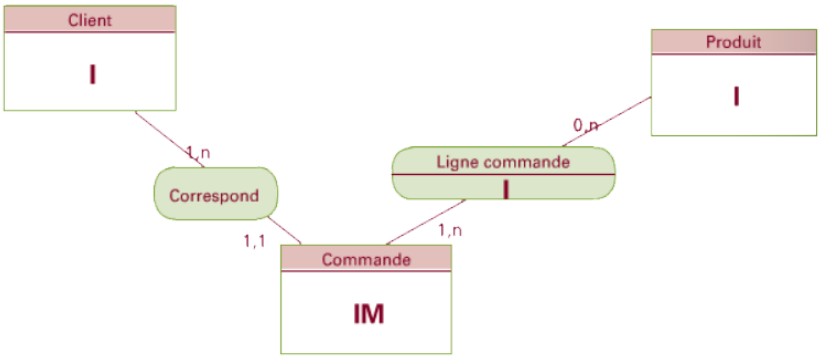
Le MOD est tournés vers la gestion des droits des utilisateurs sur une base de données. C'est une reprise directe du MCD correspondant, au sein duquel on précise les droits d'un profil d'utilisateurs (i.e. un acteur, un poste de travail, ces mêmes acteurs internes que l'on retrouve dans le modèle de flux (MF) et le modèle organisationnel des traitements (MOT et MOTA)). On rédige **un MOD par acteur interne**. Les droits sont précisés :
- soit à côté de chaque entité et association ;
- soit en lieu et place des propriétés (resp. propriétés portées) des entités (resp. association).

En BTS CGO, on retrouve également les appellations : **modèle de(s) vues** et **modèle CIMS**. Notez que, lorsqu'un programmeur parle de vue, il ne parle absolument pas de MOD… Il parlera en règle général de vue au sens de l'architecture MVC (hors programme). Cette terminologie n'est donc ni générique ni appropriée, mais à retenir.

On distingue les droits suivants :
- **Création** (*Create*) : droit d'effectuer une requête de type INSERT, sous-entendu création d'un enregistrement (d'un « tuple ») ;
- **Interrogation** (Read) : droit d'effectuer une requête de type SELECT, sous-entendu droit de lecture, de récupération d'enregistrements ;
- **Modification** (Update) : droit d'effectuer une requête de type UPDATE, sous-entendu droit de modifier un enregistrement ;
- **Suppression** (Delete) : droit d'effectuer une requête de type DELETE, sous-entendu droit de supprimer un enregistrement.

N.B. : le MOD peut se déduire du MOTA dans la mesure où l'acteur interne doit pouvoir effectuer les actions décrites dans le MOTA.

Ci-après un exemple de MOD :

Au service facturation, on peut : - consulter les informations clients (fiche client, liste des clients) ; - consulter une ou plusieurs commande ; - modifier l'état d'une commande (entité Commande) mais pas le détail (Ligne commande) ; - consulter les informations produits (fiche produit, liste des produits).	*Poste de travail :* service facturation 

2. La gestion de droits

Le MOD fait l'hypothèse que les droits des utilisateurs soient gérés au niveau base de données. Et il existe en effet des requêtes SQL (de type GRANT) permettant de spécifier des droits sur les Tables. De fait, les droits sont bien rarement gérés à ce niveau. On se contente le plus souvent de sécuriser l'accès à la base de données (un ou plusieurs comptes utilisateurs) de sorte que seuls le ou les administrateurs de la base de données puissent s'y connecter. Quant aux droits des utilisateurs, ils sont gérés « applicativement » (i.e. via une application logicielle), à savoir au niveau logiciel aux moyens de divers procédés.

Aussi, intéressons-nous à présent à des méthodes de gestion des droits plus réalistes.

2.1. Les concepts RBAC et DBAC

RBAC *(Role-Based Access Control)*

Le **contrôle d'accès par rôle** est un procédé usuel permettant de restreindre l'accès aux fonctionnalités d'un logiciel. Ces restrictions sont établies en fonction du rôle de l'utilisateur, à savoir en fonction du ou des profils de l'utilisateur (exemple : profils « utilisateur », « modérateur » et « administrateurs » sur un forum). Ces restrictions d'accès peuvent porter :
- sur des interfaces utilisateurs (des écrans utilisateurs), encore appelées IHM pour Interfaces Homme-Machine (ou encore GUI en anglais, pour *Graphical User Interface*). Autrement dit, on fait dépendre l'affichage du profil de l'utilisateur ;
- sur des ressources (fichiers et autres données) ;
- sur des traitements (éventuellement qualifiés de services). Autrement dit, s'il dispose d'un profil (rôle) spécifique, un utilisateur sera ou non autorisé à procéder à une manipulation particulière (exemple : création d'une facture, export du Fichier des Ecritures Comptables, génération de la liasse fiscale…).

Finalement, la méthode RBAC fonctionne grossièrement comme suit :
- on dispose d'utilisateurs et de profils d'utilisateurs ;
- éventuellement, chaque profil est constitué de droits ;
- lorsqu'un utilisateur s'authentifie (formulaire de connexion), des informations le concernant sont mises en session* ;
- lorsqu'un utilisateur tente d'accéder à une interface, celle-ci est potentiellement calibrée en fonction du ou des profils de l'utilisateur ;
- lorsqu'un utilisateur tente d'accéder à une ressource, on lui autorise ou refuse l'accès en fonction de son ou de ses profils ;
- lorsque l'utilisateur tente de procéder à une manipulation, on lui autorise ou refus l'accès en fonction de son ou de ses profils.

** Par session, on entend une information stockée temporaire côté serveur. Ces informations expirent passé un certain délai (le serveur les déstocke). Les sessions ne doivent pas être confondues avec les informations stockées temporairement côté client (exemple : navigateur). De telles informations ne sont plus des sessions, mais par exemple des cookies.*

DAC (Discretionary Access Control)

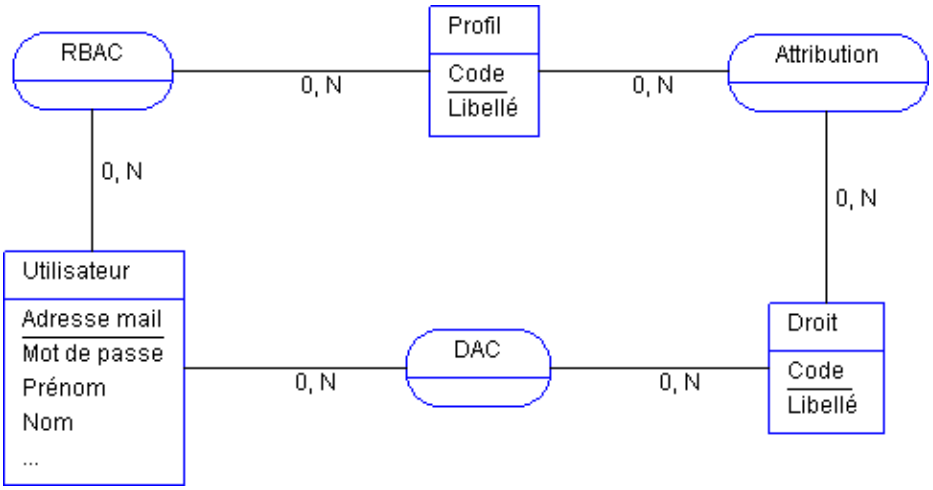
Le **contrôle d'accès discrétionnaire** vient compléter la méthode RBAC. Il permet également de restreindre l'accès aux fonctionnalités d'un logiciel. Les accès ne dépendent plus du ou des profils d'un utilisateur mais directement de l'utilisateur. A la différence de la méthode RBAC, les droits ne sont plus affectés aux profils mais à l'utilisateur, d'où l'emploi du terme discrétionnaire.

En conclusion, les méthodes RBAC et DAC fournissent un mécanisme de gestion de droits plus riche que celui décrit par les MOD, entre autre en ce qu'elles ne se cantonnent pas à la gestion de droits de création, modifi-

cation, interrogation et suppression de données dans une base de données.

2.2. Base de données utilisateurs

Afin de mettre en œuvre les méthodes RBAC et DAC, il est nécessaire de stocker les informations relatives aux utilisateurs, aux profils et aux droits. Pour ce faire, une méthode classique consiste à stocker ces informations dans une base de données utilisateurs dont on présente ci-dessous une implémentation possible (MCD) :



On remarquera que, si la mise en œuvre des restrictions peut s'avérer complexe, le stockage des informations relatives aux droits est tout à fait trivial.

Notez également, qu'en bon informaticien, **on ne stocke normalement jamais les mots de passe en clair** en base de données. En pratique, seul l'utilisateur connaît son mot de passe. **Seul le haché* est stocké** en base.

* Un haché est une « image » d'un mot de passe calculée à partir d'une fonction de hachage. Une fonction de hachage est normalement non inversible, à savoir qu'on ne peut recalculer le mot de passe d'origine (exemples de fonctions de hachage : MD5, SHA-1, SHA-256).

2.3. Annuaires LDAP

Un annuaire LDAP est un dispositif logiciel fournissant des services centralisés. Plus exactement, sur un réseau, un tel annuaire permet l'accès à des informations partagées (liste de ressources sur le réseau, liste de postes utilisateurs…). Les annuaires LDAP respectent un certain nombre de standards. En particulier, un annuaire LDAP fournit un service centralisé d'identification et d'authentification. Les annuaires LDAP les plus connus sont sans conteste l'Active Directory (Microsoft) et OpenLDAP (Libre).

En outre, c'est majoritairement au moyen d'un annuaire LDAP qu'on centralise la gestion des utilisateurs, des accès et des authentifications sur le réseau d'une organisation quelconque (exemple : établissements scolaires, entreprises…).

2.4. Services d'authentification unique

Un service d'authentification unique, appelé service SSO (Single Sign-On), est un service permettant la centralisation des authentifications. De tels services sont fréquemment utilisés au travers du web. De même qu'avec un annuaire LDAP, un service SSO consiste, pour logiciel, à déléguer l'authentification des utilisateurs à un dispositif extérieur au logiciel. Il existe divers protocoles* SSO (exemples : CAS, OAuth).