

COUCHE 7/OSI : PROTOCOLE d'APPAREIL VIRTUEL

NOTA : Le chapitre ci-dessous illustre les concepts d'appareil virtuel mais ne tient pas compte des développements récents dans ce domaine. Il est rédigé d'après une version préliminaire de la norme OSI. L'exemple fourni, terminal Vidéotex, est donné pour illustrer ces principes.

1. Applications interactives et

Applications de transfert d'information pour restitution.

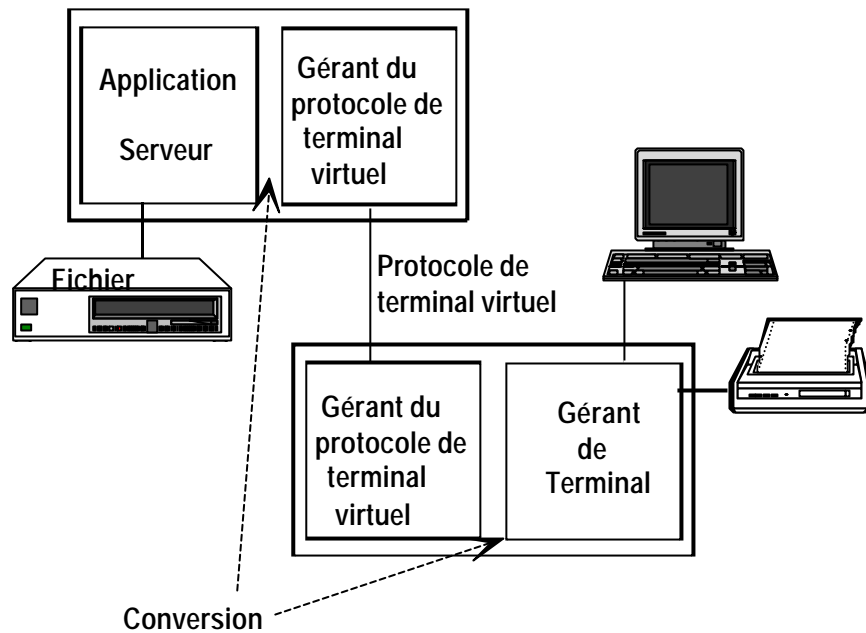
Les applications interactives sont constituées de transactions élémentaires composées chacune d'une "question" et d'une réponse. La question peut être composée en texte libre ou à l'aide d'une grille d'écran. Les échanges se font en temps réel.

Les applications de transfert d'information pour restitution s'échangent des quantités, parfois importantes, d'informations sans intervention de l'utilisateur durant le transfert. Cette information est ensuite restituée sous une forme compréhensible par l'utilisateur. Cette restitution peut se faire en temps différé quand le document est complètement transféré.

Ces deux types d'application ont en commun le problème de la restitution des données sur un terminal : écran ou imprimante. Il est nécessaire de **masquer à l'utilisateur la diversité des terminaux** et de la manière avec laquelle ils réalisent une fonction donnée. Pour cela la Présentation définit un protocole d'appareil virtuel qui décrit le langage commun (syntaxe de transfert) aux entités de Présentation indépendamment des terminaux réels ou du système de stockage des données.

Les données échangées peuvent être **textuelles** et/ou **graphiques**. On pourra utiliser la syntaxe X409 pour décrire formellement la structure des données échangées.

Le schéma ci-dessous illustre ce service d'appareil



2. Modèle conceptuel

Ce modèle conceptuel définit une représentation théorique d'un terminal, représentation commune aux entités interconnectées. Chaque application traduit ce modèle dans ses propres structures. Ce modèle fournit une représentation commune de l'image à visualiser, à stocker ou à transmettre ainsi que de l'état du terminal et des moyens de signalisation : curseur(s) ou réticule(s).

La syntaxe de transfert utilisée va permettre de décrire ce modèle conceptuel d'une manière identique pour les deux entités de présentation qui communiquent. Chaque entité d'application en aura une représentation particulière adaptée au terminal ou au système de fichiers.

2.1 Définitions

Objet atomique : Plus petite unité de données adressable séparément exemples : caractère, pixel, bit, objet graphique

Attributs : Propriétés associées à un champ de données ou un groupe de champs, éventuellement un objet atomique.

Cellule : Plus petit élément de **stockage** adressable séparément. Elle contient un objet atomique.

Image conceptuelle : Contenu d'information d'une zone de stockage de données tel qu'il est vu par les deux utilisateurs.

Aire de communication conceptuelle : CCA Zone de dépôt conceptuelle pour les deux entités qui échangent des données ou des commandes.

Zone de données conceptuelle : CDS

Sous-ensemble de la CCA où sont stockées (virtuellement) les données par les deux utilisateurs.

Chaque utilisateur doit pouvoir "visualiser" l'image conceptuelle contenue dans la CDS et consulter ou modifier les différentes zones de la CDA.

Zone de signalisation : SIDS

Sous-ensemble de la CCA où est consignée la position des moyens de signalisation : curseurs ou réticules

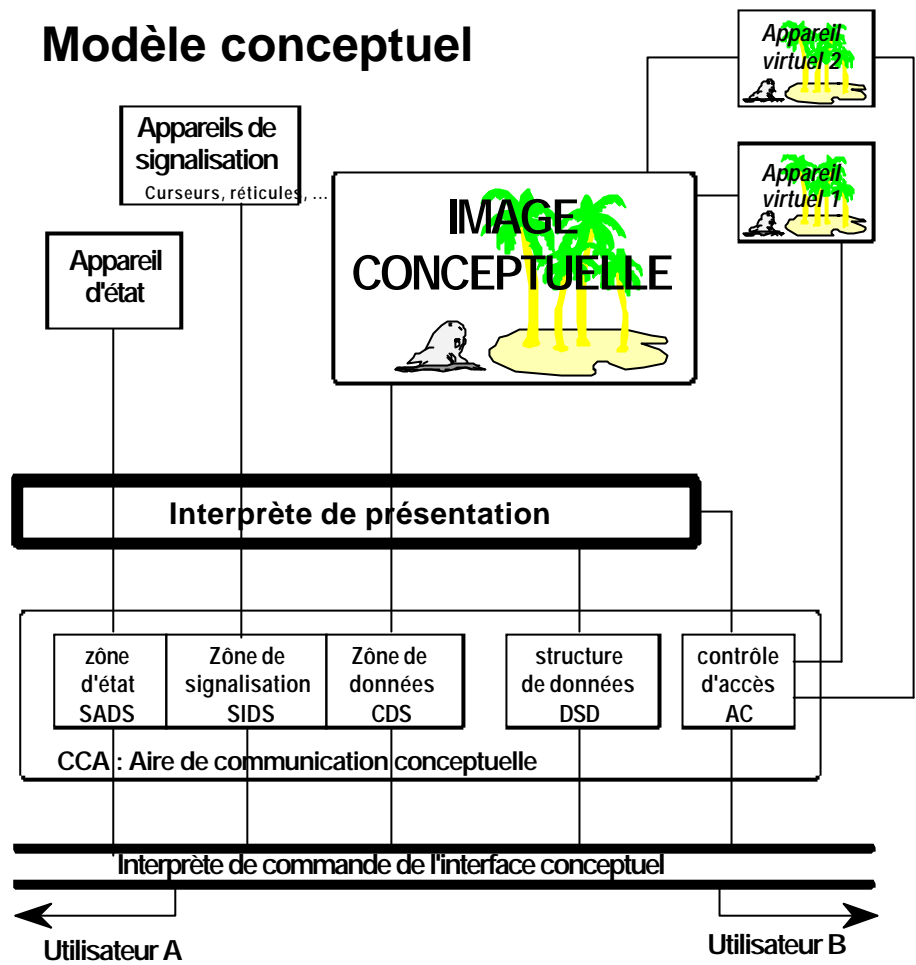
Zone d'état : SADS

Sous-ensemble de la CCA qui contient les variables d'état du terminal virtuel.

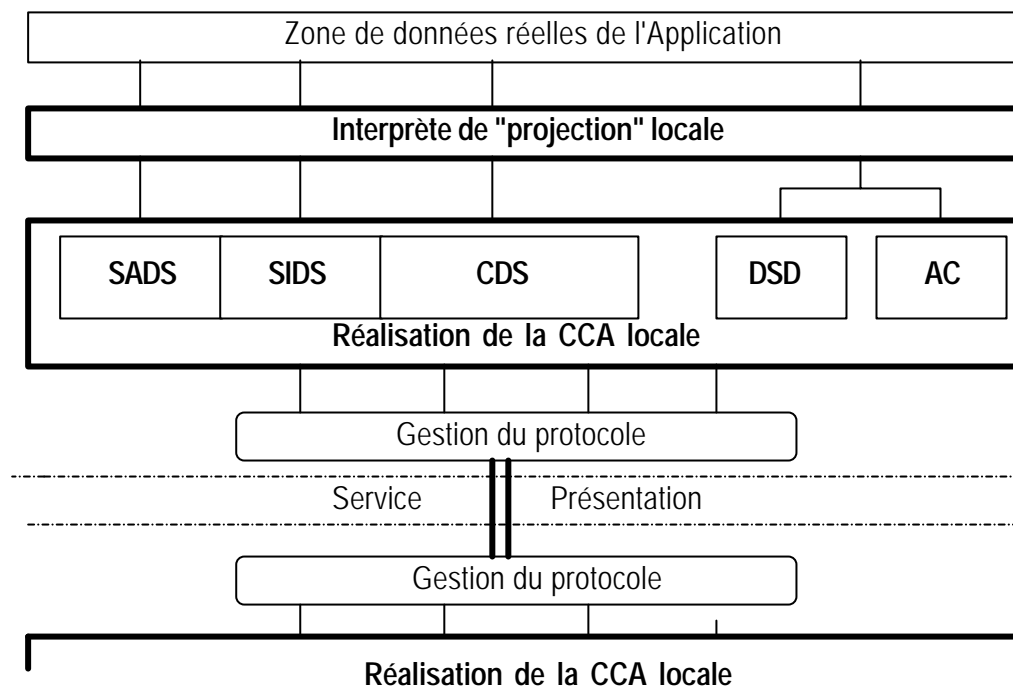
Le modèle conceptuel contient aussi des fonctions gérant le *contrôle d'accès* (AC) et un *ensemble de structures de données types* (DSD) servant de modèle de données.

2.2 Structure logique et implantation

Le schéma ci-contre illustre ce modèle conceptuel.



Pour chaque application on aura une implantation spécifique selon le schéma de principe ci-dessous :



3. Exemple ancien : Appareil virtuel Architel

L'exemple ci-dessous illustre, pour un terminal simple "**Minitel**", la notion de protocole d'appareil virtuel et de modèle conceptuel. Pour plus de détails sur ce système on se rapportera au cours sur le Vidéotex.

Pour les applications de type Vidéotex ou Télétex, l'administration française des Télécommunications a défini un protocole d'appareil virtuel à implanter dans les serveurs ou dans les concentrateurs de terminaux.

Ce protocole d'appareil s'appuie sur un service Session OSI de type BAS.

Il permet une composition des images souple et **modulaire** qui facilite la construction d'une image par assemblage de sous-images et sa modification partielle.

Les applications traitent et échangent des informations logiques sous forme de chaînes de caractères qui seront interprétées avant visualisation. L'unité logique d'information à visualiser est le **champ**.

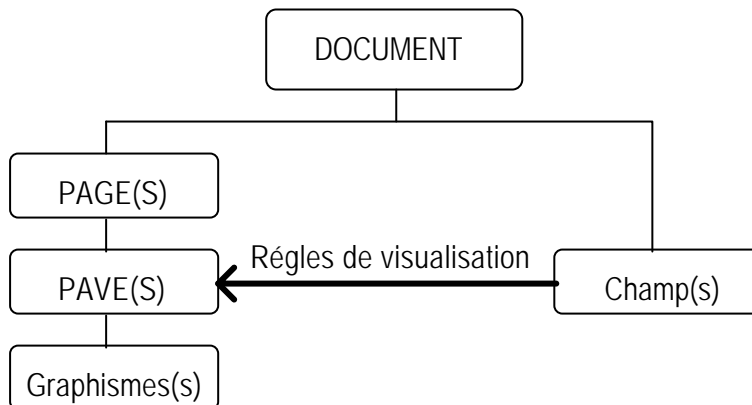
3.1 Structure de document

L'objet le plus global manipulé par le protocole est le **Document**.

L'image visualisée à un instant donné sur le terminal constitue une **page** de ce document.

Pour faciliter les traitements, les stockages ou les transferts le document est découpé en **champs**. Le champ est l'**objet de base** de la structure logique. Il est modifiable par l'opérateur en saisie ou par l'application en réponse à une question.

Dans sa structure d'images, le document est découpé en pages; chaque page est construite à partir de **pavés** qui constituent les sous-images. Un pavé est une surface de visualisation rectangulaire, aux cotés parallèles à ceux de l'écran, et de dimensions quelconques. Le contenu d'un pavé est constitué de **graphismes** qui portent l'information utile.



Rq : Un **champ** peut être **visualisé** dans un ou plusieurs **pavés**.

3.2 Signalisation

Pour manipuler un objet il faut pouvoir le désigner soit dans sa structure physique (structure d'images) soit dans sa structure logique qui permet d'échanger des informations entre utilisateur et application.

Dans la structure physique :

on utilise un **curseur** composé de quatre éléments :

- * pointeur de document
- * pointeur de page
- * pointeur de pavé
- * pointeur de graphisme

Dans la structure logique :

on utilise un **pointeur de champ**.

A un instant donné, l'application peut accéder soit à la structure d'image par le curseur, soit à la structure logique par le pointeur de champ.

3.3 Objets et modèles

Habituellement, les terminaux, simples (vidéotex), sont connectés à un concentrateur de terminaux situé à l'entrée du réseau que l'on nomme "**point d'accès**".

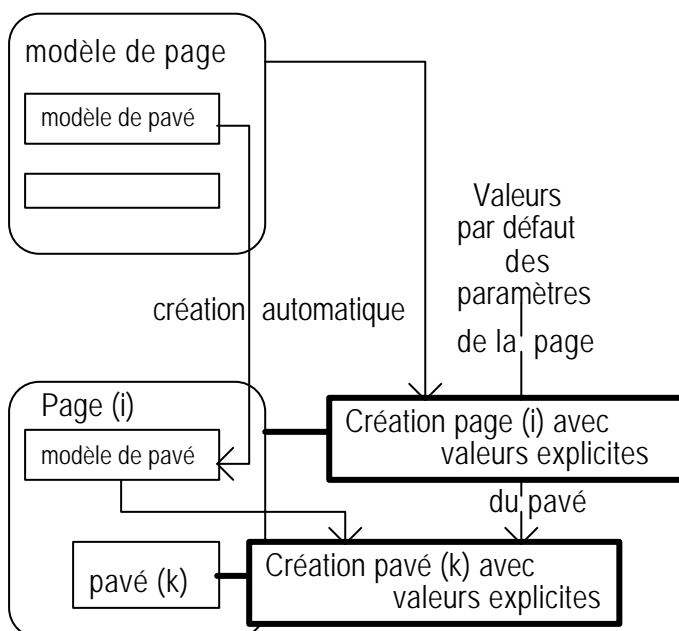
Pour diminuer les volumes de données à transférer, le point d'accès dispose d'une bibliothèque d'**objets téléchargeables** par le Serveur.

* Un tel objet n'est transféré qu'une seule fois sous forme d'éléments de protocole

* Il peut être utilisé directement ou modifié plusieurs fois par des commandes envoyées depuis le Serveur.

Chaque objet manipulable (page, pavé) comporte un ensemble de paramètres qui lui sont propres. Ainsi un pavé est caractérisé par une couleur de fond, une couleur de graphisme, un modèle de graphisme, des dimensions. L'ensemble des valeurs les plus utilisées pour les paramètres d'un objet est regroupé dans un **modèle**. Pour créer un nouvel objet on pourra copier ce modèle ou en modifier un ou plusieurs paramètres (par exemple uniquement la valeur de fond). Il faudra aussi donner une valeur explicite aux graphismes pour visualiser l'information.

Le schéma ci-dessous illustre ces mécanismes.



Un modèle de document est constitué de pages modèles.

Un modèle de page regroupe des modèles de pavés. Ainsi à la création d'un objet, on hérite des modèles de ses objets constitutifs qui sont créés automatiquement avec leurs paramètres par défaut. On pourra éventuellement les modifier.

De même, la présence d'un modèle d'objet à un niveau de structure permet de créer automatiquement les objets de niveaux supérieurs s'ils ne sont pas déjà décrits explicitement.

La modification d'un document permet d'ajouter une page en fin de liste, de modifier les éléments variables de modèle de page et le contenu des pages de document. La modification d'une page permet d'ajouter un pavé en fin de liste, modifier les éléments variables du modèle de pavé et de modifier le contenu des pavés.

La modification du contenu d'un pavé ne porte que sur le modèle de graphisme et la suite des graphismes.

La modification du contenu d'un champ ne peut porter que sur la suite de graphismes répartis dans la suite de pavés qui constituent le champ.

3.4 Programme de saisie

Pour diminuer la charge du serveur le programme de saisie est déporté au point d'accès. Il permet de rendre le traitement des informations indépendant de la conduite de la saisie sur le terminal.

L'objet de base de la structure logique étant le champ, la saisie se fait champ par champ ; chaque champ est traité par un sous-programme de saisie.

L'état global de la saisie (active ou non, rang du sous-programme en cours) est rangé dans un objet manipulable par la Présentation: le compteur de saisie.

Chaque sous-programme comporte des consignes générales et un pilote de saisie définissant les actions à effectuer selon les événements reçus (touches de fonctions). ces actions peuvent être locales ou provoquer le transfert d'information vers le serveur. Le programme de saisie devient alors inactif. (ainsi le transfert est semi-duplex :TWA au niveau Session).

3.5 Exemple d'utilisation du protocole

Le schéma donné sur la page suivante décrit un scénario du protocole. Ce scénario montre la création de pavés et leur visualisation, puis la création, la saisie et le transfert de champs.

