

Etude et conception d'une interface Web avec les SGBDs relationnels

Karim OUAHES

*Laboratoire des Logiciels de Base CE.R.I.S.T
Rue des 3 frères Aïssou, Ben Aknoun, Alger, Algérie*

Fax : 213 (2) 91.21.26 - Tél. : 213 (2) 91.18.21

E-mail : Ouahes@tassili.cerist.dz

La fin de ce siècle est marquée par les progrès importants réalisés dans le domaine de l'informatique et des télécommunications. Le mariage réussi de ces deux domaines a donné naissance aux réseaux informatiques. Le développement de ces réseaux a généré une valeur ajoutée à l'information.

Au sein de l'entreprise, la communication est devenue un moyen primaire pour diffuser et disposer au moment opportun de l'information aussi bien au niveau interne qu'externe et ce pour une gestion efficace et efficiente.

C'est par des besoins concrets se traduisant en termes de besoins d'infrastructures de communication (ex : accéder au réseau) ou de production (créer des contenus adaptées à ces moyens de diffusion), que les entreprises prennent aujourd'hui conscience de la nécessaire composante d'internet dans les échanges d'informations, entrant, sortant ou simplement circulant dans l'entreprise et qu'un modèle particulier pour des besoins spécifiques génère aujourd'hui la réflexion autour d'internet.

Initialement destiné à la communauté des chercheurs et aux universitaires, Internet est aujourd'hui utilisé par une vaste population manifestant un engouement certain pour ce réseau depuis les années 1990 et ce grâce au World-Wide Web (Web).

En plus des autres services de base d'Internet, le Web a largement contribué à l'émergence de ce réseau qui est devenu un phénomène de société, grâce à sa simplicité, sa convivialité et au principe de la navigation.

Les possibilités de communication à l'échelle planétaire offertes par le réseau Internet en font un moyen de diffusion d'une masse importante d'informations et un espace de communication, de culture et de liberté. Cette ouverture, qui permet à des utilisateurs dispersés à travers le monde de dialoguer entre eux, a induit l'idée de faire d'Internet un

vecteur dans des domaines divers pour satisfaire de nouveaux besoins exprimés tels que le télécommerce, le leletourisme ou la télémedecine.

Ces types de traitements conjugués aux mécanismes mis en oeuvre par le Web posent de façon générale le problème d'accès aux bases de données à travers le Web qui reste le service le plus attrayant du réseau Internet.

En effet, le langage utilisé par le Web est le langage HTML. C'est un langage de présentation non encore mature qui présente des limites certaines dans le domaine transactionnel.

L'accès aux bases de données requiert presque toujours la réalisation d'un élément extérieur pour servir d'interface entre le Web et une base de données existante.

I. PROBLEMATIQUE

Les possibilités de navigation à travers les différents serveurs repartis à travers le monde ont fait apparaître de nouveaux besoins en plus de ceux jusqu'ici offerts par le Web.

Ce service qui fonctionne selon le modèle client serveur est basé sur le protocole HyperText Transmission Protocol (HTTP), le langage de programmation défini pour le Web est HyperText Markup Language (HTML) qui permet une structuration et une description de pages.

Son niveau actuel ne permet cependant pas tous les types de représentation de l'information tels que l'on a l'habitude de les trouver dans les applications informatiques.

Il permet à un utilisateur de naviguer à travers des pages mises à disposition dans un serveur par un auteur grâce à des liens de type hypertexte. (On est donc dans un monde de consultation où on a simplement un choix dans le cheminement parmi les pages, sans aucune interactivité réelle entre l'utilisateur et le système qui fournit les documents. In effet, les faiblesses avérées du langage HTML dans le domaine des traitements interactifs rendent difficile la satisfaction des nouveaux besoins exprimés, le problème d'accès aux bases de données se pose de façon générale pour la réalisation de documents dynamiques c'est-à-dire des documents générés au moment de la demande de l'utilisateur.

C'est ainsi que se pose le besoin de réaliser une interface pour permettre la rencontre entre le Web et les divers systèmes de gestion de base de données.

II. ETUDE DE L'INTERFACE

II.1. Le Common Gateway Interface (CGI)

L'utilisation la plus courante du service Web est la navigation transparente à travers les différents serveurs d'un réseau Internet. Cette navigation est possible grâce à des liens (URLs) qui permettent l'affichage successif de pages HTML. Ces dernières sont très

souvent statiques (elles sont créées par un auteur et mises à disposition pour consultation). Des opérations demandant l'accès à des bases de données nécessitent une interaction entre l'utilisateur et le serveur Web.

Pour la création de tels documents dynamiques (des documents générés sur demande de l'utilisateur) les formulaires constituent la réponse apportée par le langage HTML.

Les formulaires ou forms sont des fiches qui permettent à l'utilisateur grâce à des balises HTML, de remplir des zones de saisie.

Chacune de ces zones comporte un nom symbolique et un champ de saisie. Le lien du remplissage du formulaire est déclenché lorsque l'on appuie sur un bouton spécial appelé "bouton de soumission". Cette fin provoque l'envoi des données saisies qui requièrent pour leur traitement un élément approprié qui est presque toujours un programme ou script s'exécutant dans un espace du serveur réservé à cet effet : le Common Gateway Interface (CGI).

Ce programme chargé du traitement du formulaire reçoit des couples constitués de l'identificateur (le nom symbolique) de chaque zone et de sa valeur saisie (qui peut être nulle).

L'interface permet ensuite d'établir le dialogue entre un browser soumettant son formulaire et un autre programme accessible depuis la machine serveur comme un système de gestion de base de données par exemple. Le serveur Web fonctionne ainsi comme un client du SGBD.

Le Web utilise CGI (Common Gateway Interface) qui est un standard pour l'écriture de passerelles entre des serveurs d'informations tel que HTTP et des programmes externes.

L'utilisation de CGI permet entre autres possibilités d'utiliser dans un programme les variables d'environnement qui sont d'une grande importance. En effet, elles permettent l'accès à de nombreuses informations utiles dans la programmation CGI notamment sur le serveur, le client et la machine qui exécute le client .

Il est à noter toutefois que l'offre Microsoft pour Internet inclut l'interface MISAPI (Microsoft Internet Server Application Program Interface).

MISAPI pour Windows NT peut être utilisé pour écrire des programmes tournant sur un serveur Web dans la plupart des langages 32 bits. Ces programmes compilés sous forme de DLLs sont chargés par le service à son démarrage. Comme l'interface CGI, MISAPI permet de mettre en oeuvre des formulaires dont les informations peuvent être récupérées et traitées par les programmes externes.

De plus, Microsoft offre le package Internet DataBase Gateway (IDG) qui est une passerelle pour interfacier les documents Web avec les bases de données. Cette interface est basée sur la technique ODBC; elle fonctionne avec la plupart des SGBDs.

D'autre part, IDC (Internet Database Connector) permet l'accès aux bases de données.

II.2. Alternatives de réalisation de l'interface

Les alternatives de réalisation de l'interface mettent presque toutes en oeuvre un élément externe. On peut en distinguer trois.

La façon la plus intuitive consiste à l'écriture de scripts CGI. Cette approche consiste en l'utilisation de formulaires qui permettent d'établir une zone de dialogue entre l'utilisateur et le serveur; les traitements nécessaires étant réalisés par les scripts. Par l'utilisation de formulaires HTML, le dialogue dans ce cas est géré à travers la passerelle standard Common Gateway Interface (CGI).

Une deuxième alternative pour réaliser l'interface est possible par l'utilisation d'un élément standard qui joue le rôle de passerelle entre les applications Web et le système de gestion de bases de données.

Enfin, une troisième alternative pourrait consister en une extension du langage HTML pour héberger directement dans les documents HTML des sections qui permettent d'établir une interactivité entre l'utilisateur et le serveur.

II.2.1. Présentation des différentes alternatives

11.2.1.1. Première alternative

Elle consiste en l'utilisation de langages de programmation permettant des extensions SQL. Comme énoncé plus haut, la réalisation d'une interface Web avec un SGBD relationnel par l'utilisation de scripts met en oeuvre deux parues principales : les formulaires et les programmes ou scripts CGI.

Cette méthode est basée sur la programmation CGI qui vise à construire le document HTML correspondant à un lien hypertexte au moment même où l'on active ce lien. Le document n'est jamais stocké dans un fichier et est envoyé au client au fur et à mesure de sa construction grâce aux liens exécutables. Ces liens identifient des programmes à exécuter dont la sortie standard est envoyée au client.

Ces programmes lancés à partir de liens exécutables sont appelés des scripts CGI.

11.2.1.1.1. Les formulaires

Ils permettent de passer des arguments à un programme et gèrent les liens exécutables. Leur création dans un document HTML se fait par l'utilisation de la commande `<F()RM>` dont la syntaxe est :

<F()RMACTION= "URL " METHOD=/CET \ POST]>....<IF()RM>

où :

URL désigne le programme chargé de réaliser le traitement;
METHOD le mode de passation des paramètres.

Entre <FORM> et </FORM>, on peut définir différents types de champs :

- des champs d'entrée simple par *INPUT*;
- des boutons et des menus de sélection par la balise *SELECT*.
- des zones d'entrée de texte etc.;

Suivant son type, un champ peut être défini par :

- un nom;
- une valeur;
- une taille etc;

11.2.1.1.2. Les scripts

Ce sont des éléments externes au langage HTML. Ils constituent des programmes dont le rôle est de servir de passerelle entre l'utilisateur qui soumet ses données grâce au formulaire et le système de gestion de base de données.

Son écriture consiste, pour le développeur, à réaliser intégralement le programme CGI qui comprend tout le code SQL nécessaire à ses traitements. Le principe de fonctionnement est le suivant :

- le script récupère les paramètres transmis par le formulaire;
- il envoie une requête SQL au serveur;
- le serveur exécute le programme indiqué et envoie les résultats au script sous forme brute;
- le script récupère les résultats puis les met en forme en ajoutant du code HTML.

Tout langage capable d'avoir une sortie standard peut être utilisé pour écrire des scripts CGI.

- Exemple :
- C, Tel, Fortran, Perl, tous les shells UNIX pour les serveurs UNIX et VMS;
 - Visual C ++, Visual Basic sur PC-Windows ;
 - Pro*c, Oraperl pour Oracle sous UNIX.
 - ESQ-L-C pour Informix et SQL Server;

11.2.1.1.3. Méthode de communication

Le dialogue entre les formulaires et les scripts se fait à travers l'interface standard de programmation CGI. Cette passerelle entre un serveur Web et une application permet aux formulaires de communiquer des informations aux programmes grâce à des variables d'environnement. Les deux qui sont les plus utilisées sont : *QUERY_STRING* et *PATH INFO*.

L'importance de ces deux variables est surtout marquée par le fait que la syntaxe de l'URL n'offre pas de possibilité de passer des paramètres au script comme le permettrait une ligne de commande. Biles **constituent** un moyen d'indiquer, en plus du nom du script à exécuter, d'autres informations.

La programmation CGI appliquée au formulaire doit permettre d'envoyer au script l'ensemble des données saisies par l'utilisateur. Afin de transmettre ces données au serveur, le client construit une chaîne de caractères contenant l'ensemble des couples (nom du champ, valeur saisie).

Cette chaîne de caractères, accessible par la suite au niveau du script, est codée sous le format suivant :

nom_champ1=valeurl& nom_champ2~\nlcur2& . . . & nom_champn~valeurn

les champs n'ayant pas été saisis recevant la chaîne vide connue valeur par défaut et les champs à valeurs multiples apparaissant avec autant de couples que de valeurs sélectionnées.

Dans cette chaîne, les espaces sont remplacés par des '+' et les caractères spéciaux par leur valeur hexadécimale sous le format '%xx\'. On parle de chaîne URL-encodée.

La transmission de cette chaîne de caractères au serveur est faite à la fin de la saisie du formulaire. Cette fin est matérialisée lorsqu'on appuie sur le bouton de soumission (*SUBMIT BUTTON*).

Le mécanisme de transport de ces données est fonction de la méthode indiquée par l'attribut *METHOD* de la balise *<I()RM>*. On distingue deux méthodes.

- la méthode CET

Elle ajoute la chaîne de caractères à l'URL référençant le script à exécuter: les informations étant disponibles grâce à la variable d'environnement *QUERY_STRING*

- la méthode POST

Elle utilise une séquence spéciale HTTP. Contrairement à la méthode GET, les informations issues du formulaire sont transmises à travers l'unité standard *stdin*. Le nombre de caractères à lire est dans ce cas contenu dans la variable d'environnement *CONTENT_LENGTH*.

L'ouverture du système induite par l'exécution de script sur le serveur pose le problème de sécurité et de confidentialité. Ainsi la méthode POST reste la plus utilisée pour le traitement des formulaires. En effet, elle permet de masquer les données transmises alors qu'avec la méthode GET, elles sont accolées à l'URL et affichées par la plupart des browsers.

Après le traitement des informations issues du formulaire, le script a en charge la

11.2.1.2.1. Les formulaires

Comme avec la première alternative, les formulaires permettent de passer des arguments aux programmes. Leur création est réalisée selon la méthode exposée précédemment. Il reste que dans l'approche ORACLE, les formulaires peuvent être réalisés aussi grâce aux procédures et fonctions du script standard.

11.2.1.2.2. Les scripts

Ce sont des programmes qui réalisent les traitements. Ils sont écrits en PL/SQL et compilés sous forme de procédures stockées. Leur rôle dans cette approche est :

- de récupérer les données issues des formulaires;
- de les traiter;
- de récupérer les résultats des requêtes SQL dont le formatage en HTML est réalisé par le script standard grâce à des fonctions et procédures PL/SQL d'encapsulation en HTML.

11.2.1.2.3. Le script standard

Une fois installé, il joue le rôle d'interface entre le serveur Web et le SGBD. Il se charge de lancer les procédures sur le serveur avec les paramètres adéquats et de récupérer les données déjà formatées que lui renvoie le serveur.

Il renferme un ensemble de fonctions et procédures stockées PL/SQL qui reprennent la plupart des fonctionnalités du langage HTML notamment la création des formulaires et de leurs composants et le formatage des tables.

II.2.1.2.4. La méthode de communication

Le dialogue entre les formulaires et le script standard est réalisé à travers l'interface CGI. Le mécanisme de transport peut dans cette approche aussi être basé sur les méthodes GET et POST exposées précédemment; le nom de la procédure à exécuter sur le serveur étant récupéré par le script standard grâce à la variable d'environnement PATHINFO de l'URL.

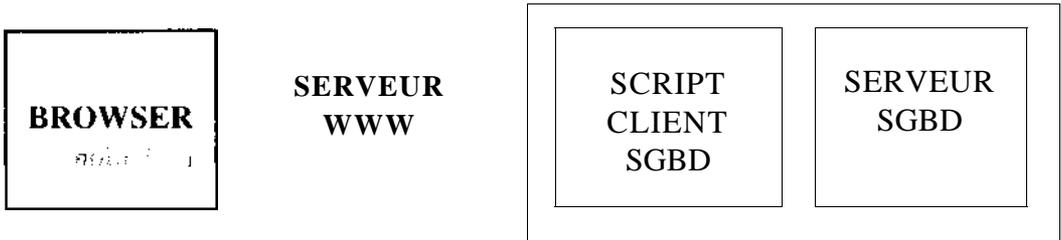
Les paramètres de cette dernière procédure sont les valeurs des champs de saisie du formulaire. Ils sont passés au script par la variable QULRY_STRING pour la méthode POST et CONTENTLENGTH pour la méthode GET.

Le format de la chaîne de caractères construite avec les données du formulaire reste identique à celui exposé précédemment. Toutefois, l'appel d'une procédure stockée devra satisfaire certaines contraintes:

- même nombre de champs de saisie dans le formulaire que de paramètres de la procédure;
- identité entre les noms de champs à saisir et ceux des paramètres;

- unicité des noms de champs de saisie dans le formulaire.

Le schéma suivant illustre les mécanismes mis en oeuvre par l'approche ORACLE.



- . Requête sur l'URL
- . Lancement du script standard
- © Appel à la procédure indiquée au script
- Résultats formatés en HTMLSGBD
- . envoi des résultats
- . envoi des résultats

11,2.1.3. Troisième alternative

Elle est axée sur une extension du langage HTML pour établir le dialogue entre le serveur et l'utilisateur. En effet, un document HTML est structuré par l'utilisation de commandes spécifiques appelées balises. Ces balises sont interprétées par les browsers lors de l'exploitation du document HTML.

La réalisation d'une interface par une extension du langage HTML consiste à inclure dans une page HTML de nouvelles sections. Ces sections permettraient d'intégrer dans un document HTML des divisions qui, lors de l'interprétation par le client Web du document HTML, demanderont le lancement de l'outil de développement du code pour le compiler éventuellement et l'exécuter. Cette exécution devrait pouvoir permettre la gestion de pages Web interactives.

Cette méthode est mise en oeuvre dans les approches de Microsoft et INFORMIX.

II.2.1.3.1. L'approche MICROSOFT

Elle est basée sur l'utilisation de Visual Basic Script qui est un sous-ensemble du langage Visual Basic. Visual Basic tire partie des caractéristiques des browsers et des serveurs Web.

L'approche Microsoft consiste à inclure dans un document HTML du code Visual Basic Script grâce à des balises spécifiques. Ce code est représenté comme un texte ASCII. Il est compilé au moment du chargement du document le contenant. L'interprétation de la section définie entre le tag de début et le tag de fin est faite grâce à Visual Basic Script qui compile et exécute le code.

H.2.1.3.2. L'approche INFORMIX

Comme celle de Microsoft, l'approche Informix est basée sur l'extension du langage HTML. Elle consiste, grâce à une balise spécifique, à intégrer directement des requêtes SQL et les instructions de formatage des résultats de ces requêtes dans des pages HTML qui sont stockées dans une base de données de type relationnel objet.

Cette approche supprime la nécessité de développer une application CGI pour accéder dynamiquement aux données des SGBDs. Pour ce faire, elle met en oeuvre:

- des fonctions qui permettent d'extraire des informations à partir de pages Web et de récupérer des variables d'environnement CGI;
- des balises qui permettent aux instructions SQL de s'exécuter dynamiquement dans les pages HTML.

La méthode Informix utilise deux principaux composants :

- le pilote CGI Webdriver

Il permet de se connecter à une base de données Illustra de type relationnel objet. Il extrait les pages HTML et retourne le résultat HTML au serveur Web grâce à la fonction d'extraction WebExplode.

- la fonction WebExplode

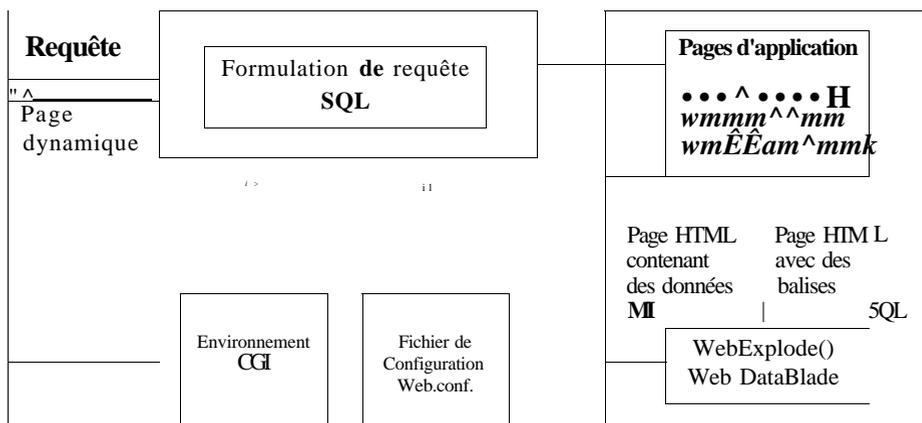
Elle décode les données reçues et construit dynamiquement les instructions SQL contenues dans les pages HTML. Elle se charge ensuite de leur exécution et retourne le résultat au pilote CGI.

Les mécanismes de cette approche basée sur l'utilisation de bases de données de type relationnel objet peuvent être représentés selon le schéma suivant:

Serveur Web (hHpd)

Pilote CGI (Webdriver)

SGBDRO Illustra



Le principe de fonctionnement de cette approche est le suivant :

- le browser soumet une requête pour invoquer le pilote CGI;
- le pilote CGI Webdriver construit dynamiquement les requêtes SQL sur la base des informations fournies par le browser pour récupérer la page d'application grâce à la fonction WebExplode;
- WebExplode récupère la page de la base de données Illustra, exécute les instructions SQL contenues dans la page et formate les résultats;
- les résultats sont envoyés ensuite au client par l'intermédiaire du serveur Web par le pilote CGI.

Le dialogue entre le client et le serveur Web est ainsi géré à travers l'application CGI Webdriver qui réalise une interface avec la fonction d'extraction de pages, de construction dynamique et d'exécution des requêtes et de récupération des résultats. Cette interface exploite un ensemble de données qui peuvent provenir :

- du formulaire par l'intermédiaire des champs de saisie ;
- des variables d'environnement PATHINFO ou QUERYSTRING selon la méthode de transport;
- un fichier de configuration qui spécifie les valeurs par défaut pour l'interface CGI Webdriver.

La réalisation de l'interface Web avec les SGBDs dans les différentes approches est faite grâce à la rencontre de deux technologies client/serveur. En effet, le browser est un client

du serveur Web qui est lui même client du serveur de base de données. Deux configurations peuvent être envisagées pour permettre au serveur Web de fonctionner comme client du SGBD.

- Le serveur Web et le SGBD sont sur la même machine : dans ce cas les besoins en logiciels se résument aux produits de développement distribués avec le SGBD (des API, des bibliothèques etc.) permettant de manipuler le langage de requête SQL.
- Le serveur Web et le SGBD sont sur deux machines différentes : en plus des produits ci-dessus mentionnés, on a besoin de la couche réseau du SGBD pour gérer le dialogue entre les deux machines.

11.2.2. Comparaison des différentes approches

Les différentes méthodes analysées mettent toutes en oeuvre des formulaires. Elles permettent dans leur ensemble de s'affranchir du type de système de gestion de base de données par une technique telle que ODBC offrant ainsi la possibilité de leur mise en oeuvre dans les divers types d'applications qui supportent cette interface.

De plus, elles sont toutes compatibles CGI mais requièrent pour la réalisation de l'interface Web avec les SGBDs relationnels un ou plusieurs éléments externes de mise en oeuvre différente.

11.2.2.1. L'utilisation de langages de programmation classique

Dans cette approche, le développement est réalisé sur le serveur Web et toutes les commandes SQL sont incluses dans le code du script CGI.

Le développement d'une interface Web avec les SGBDs relationnels requiert dès lors l'utilisation de plusieurs langages. Ce qui rend les applications moins lisibles et leur maintenance plus difficile.

11.2.2.2. L'approche ORACLE

Elle représente historiquement la première méthode à mettre en oeuvre une interface standard entre le serveur Web et une base de données.

Avec cette approche, le développement s'ait uniquement sur le serveur ORACLE à l'aide d'un seul langage facilite la maintenance des applications, les formulaires pouvant aussi être réalisés avec ce même langage. Toutefois cette approche présente une limite liée à la spécificité du langage PL/SQL aux bases de données ORACLE.

11.2.2.3. L'approche MICROSOFT

Elle met en oeuvre un seul langage de programmation Visual Basic Script qui supporte la technique OLE. Son intégration avec l'offre globale logicielle de Microsoft lui offre, entre autres, la possibilité d'utiliser le browser Internet Explorer.

Ce navigateur qui présentera l'interface OLE automation permettra avec cette approche d'exécuter d'autres applications.

II.2.2.4. L'approche INFORMIX

Basée sur un système de gestion de base de données relationnel objet, cette approche fournit la puissance des objets et la flexibilité du relationnel grâce au serveur Illustra. Elle supprime la nécessité de programmation CGI et offre la possibilité de gestion de données de type simple (caractère, alphanumérique, texte) et complexe (image, son, vidéo) dans une même base de données. Cette caractéristique de l'approche Informix constitue un apport non négligeable dans le domaine du Web.

De plus, l'intégration du code SOL dans les pages HTML offre une plus grande lisibilité des applications.

II.2.3. Approches testées

II.2.3.1. Approche ESQ-L-C

Cette première méthode a consisté à mettre en oeuvre des outils spécifiques au SGBD INFORMIX. En effet INFORMIX offre ESQ-L-C qui, basé sur le langage C, permet des extensions pour réaliser les opérations de :

- connexion à une base de données INFORMIX;
- formulation et d'exécution de requêtes SQL;
- et de traitement des résultats.

H.2.3.2. Approche ODBC

ODBC est une interface pour accéder aux bases de données. C'est une technique de Microsoft qui définit une méthode d'accès à des systèmes de gestion de bases de données multiples à travers une seule interface de programmation. L'interface ODBC définit des appels de fonctions qui permettent à une application de se connecter à un SGBD, d'exécuter des instructions SQL et de récupérer des résultats.

Ces opérations sont supportées par un fichier DLL (le driver manager) qui se charge de l'adaptation des fonctions au pilote ODBC spécifique au SGBD utilisé. De plus, avec la technique ODBC une application cliente n'a pas besoin de savoir la localisation exacte de la base de données (qui peut être locale ou distante), le mode de communication utilisé pour accéder à la base de données ou le type même de SGBD. ODBC requiert tout simplement que les opérations d'accès à une base de données soient spécifiées en SQL.

II.2.3.3. Approche DB-Library

DB-Library est un ensemble de fonctions et de macros G qui permettent à une application d'accéder et d'interagir avec SQL Server. DB-Library offre un ensemble varié d'interfaces de programmation d'applications (APIs) pour l'ouverture de connexion à SQL Server, la formulation de requête, la soumission de requête en batch au serveur, l'exécution de procédures stockées sur des serveurs distants et la récupération de

résultats. Ces APIs permettent des opérations de chargement de données à partir de fichiers ou de variables vers un serveur et vice-versa.

DB-Library est une interface de programmation d'applications base de données destinée à SQL Server. Toutefois, grâce à l'interface Open Data Services, DB-Library peut être utilisée pour accéder à tout type de serveur de base de données au moyen de passerelles appropriées. En outre, elle offre des possibilités de formulation dynamique de requêtes SQL, de récupérer et de formater les résultats, elle est basée sur :

- la création de formulaires HTML par un éditeur spécialisé;
- et l'écriture des scripts en C pour réaliser les traitements.

Pour la réalisation de l'interface, les formulaires permettent à l'utilisateur de soumettre des données. Leur traitement est ensuite réalisé par les scripts selon les étapes :

- de récupération des données issues du formulaire;
- de formulation des requêtes en fonction du type de traitement;
- de récupération et de formatage des résultats avant leur transmission.

Les données issues du formulaire sont transmises sous forme codée. Elles constituent une chaîne de caractères formée de l'ensemble des couples (nom de champ, valeur saisie). Les couples sont limités par le signe '&'; chaque nom de champ étant séparé de sa valeur correspondante par le signe '='.

Dans cette chaîne, les caractères spéciaux sont remplacés par leur valeur hexadécimale sous la forme '%xx' et les espaces par des signes '+'. L'étape de récupération des données du formulaire consiste alors à :

- transformer les signes '+' en espaces;
- convertir chaque valeur hexadécimale en caractère spécial;
- éclater la chaîne de caractères en couples (nom de champ, valeur saisie);
- et séparer les couples par affectation des valeurs aux noms de champ.

La formulation dynamique des requêtes SQL est ensuite faite par exploitation directe de ces résultats. Leur soumission, leur exécution et la récupération de leurs résultats sont réalisées grâce aux interfaces de programmation d'application (APIs).

Les fonctions d'affichage du langage C permettent enfin de formater les résultats avant leur transmission. Pour assurer leur bonne interprétation par le client, cette étape d'envoi des résultats devra toutefois être précédée d'un prologue pour préciser le type de données à transmettre.

III. CONCLUSION

Dans les différents domaines de la vie active, il est acquis par tout le monde qu'une gestion efficace des affaires passe par une bonne politique d'information. La diffusion et la récolte de l'information s'imposent dès lors aussi bien au plan interne qu'externe. La communication est devenue un concept à la mode, ce qui confère à un réseau tel qu'Internet une place de choix dans la vie des entreprises mais aussi des hommes. Ses possibilités de communication à l'échelle planétaire, la simplicité et la convivialité du Web ont suscité un engouement populaire certain et fait naître des besoins nouveaux tels que le télécommerce, le téléenseignement, le télé tourisme, la télé médecine. Certains de ces besoins nécessitent, pour leur satisfaction, des traitements de type transactionnel; c'est le cas notamment du télécommerce ou commerce à distance. Cette catégorie de besoin pose au niveau du service Web de façon générale le problème d'accès aux bases de données et font de l'interface Web avec les SGBDs un sujet d'actualité.

Les méthodes existantes pour la résolution de ce problème nécessitent presque toutes l'utilisation de formulaires et un élément externe au langage HTML du Web. Elles peuvent être mises en oeuvre par :

- l'utilisation de langages de programmation classiques qui offrent des possibilités d'extensions SQL et une sortie standard;
- la réalisation d'un script standard qui joue le rôle d'interface avec une base de données;
- une extension du langage HTML qui permet d'insérer directement dans une page HTML des codes qui permettent une interactivité entre l'utilisateur et le serveur.

Ces programmes ou encore scripts utilisent les interfaces de programmation Internet Database Gateway (IDG) de Microsoft et (ommon Gateway Interface (CGI) qui permettent l'intercommunication entre les formulaires et les scripts.

Pour ces méthodes, une technique telle que ODBC de Microsoft permet de s'affranchir du type de système de gestion de bases de données. Toutefois, les mécanismes du langage HTML et du protocole HTTP rendent délicate la réalisation de certaines opérations sur les bases de données telles que l'insertion, la modification et la suppression.

Toutefois, il est à noter qu'un langage tel que JAVA qui s'impose de plus en plus comme un standard pourrait, grâce aux capacités offertes de réalisation de certains traitements locaux aux browsers, ouvrir des perspectives pour la résolution de ces problèmes.

Il reste que la réalisation d'une interface Web avec les SGBDs relationnels (est un sujet d'actualité.

Références Bibliographiques

1) HTML et la programmation de serveurs Web

CHALEAT Philippe
CHARNAY Daniel Editions EYROLLES 1996

2) Le World-Wide Web

DAGORN François
CROSS Claude

3) INFORMIX-ESQL/C for DOS

Programmer's Manual Version 4.1

4) CD-ROM Microsoft ïchnet

Février, Mars, Avril, Mai L996

- Visual Basic Script : Visual Basic comes to the Net;
- Visual Basic Script : Working description.
- Creating Web applications for the Microsoft Internet Information Server.
- A high-level look at Microsoft Internet Information Server.
- ODBC Drivers.
- Delving Deeper into ODBC.
- The ODBC data types.
- MS Open DataBase Connectivity (ODBC) : Status and Future Overview.
- Open DataBase Connectivity (ODBC);
- The ODBC.solution;
- ODBC and DB-Library : A comparison of the APIs;
- Writing Interoperable Applications with ODBC;
- ODBC Technical Overview.

5) NCSA HTTPd overview

- <http://www.w3.org/h...ietf-http-spec.html>

6) The Common Gateway Interface

<http://hoohoo.ncsa.uiuc.edu/cgi/intro.html>
<http://hoohoo.ncsa.uiuc.edu/primer.html>

7) CGI : FORMS

<http://hoohoo.ncsa.uiuc.edu/cgi/formis.html>

8) The Common Gateway Interface : Spécification

<http://hoohoo.ncsa.uiuc.edu/cgi/interface.html>

9) A WEB to database Interface

<http://www.dtv.dk/~bfr/wdv/>

10) CGI Environment variables

<http://hoohoo.ncsa.uiuc.edu/cgi/env.html>

11) CGI Command Line options

<http://hoohoo.ncsa.uiuc.edu/cgi/cl.html>

12) CGI script Input

<http://hoohoo.ncsa.uiuc.edu/cgi/in.html>

13) Web DataBlade Users Guide

Illustra Information Technologies. Inc.