

COMMENT J'AI COMMENCE A TRAVAIL..., M'AMUSER AVEC APL.

par Bernard SECRET

En 1978, la production d'un budget et son analyse ultérieure, pour une entreprise fabriquant et commercialisant des produits de grande consommation, nécessitait, pour mes collaboratrices et moi, d'accomplir, entre autres, de nombreuses tâches fastidieuses telles que calculs manuels répétitifs, dactylographie des états et leur vérification au moyen d'une machine à calculer.

Celles-ci étaient à recommencer à chaque nouvelle hypothèse et l'ampleur des travaux avait souvent tendance à en limiter le nombre et à compromettre, parfois, la critique des résultats obtenus.

Comme il était devenu patent que le service informatique de notre société ne pourrait prendre en charge, dans un avenir prévisible, l'informatisation de nos travaux, nous recherchions depuis quelques mois, un système qui, en réunissant une machine à écrire et une machine à calculer, nous soulagerait, au moins, pour les travaux de publication des versions définitives. Effectivement cette solution électromécanique était expérimentée par un distributeur, pendant que certains autres proposaient déjà des matériels électroniques, mais d'une taille imposante et dont la mise en œuvre n'était pas toujours évidente.

Nous commencions à désespérer de trouver, en dehors du domaine informatique classique d'alors, le système théoriquement simple, pratique, peu encombrant et raisonnablement économique. Alors que nous pensions à tort, que la taille ne nos problèmes ne devait pas l'intéresser, nous nous sommes décidés un jour, à en soumettre une maquette à la société IBM.

C'est ainsi qu'un vendredi en fin d'après-midi, un de ses ingénieurs commerciaux me proposait de traiter moi-même mes calculs sur un de leurs ordinateurs, qu'il pouvait d'ailleurs me confier pour le week-end et la semaine suivante.

Je découvrais ainsi l'ordinateur portable (IBM 5100) et l'APL. Ce fût un coup de foudre !

Il s'agissait, et c'était pour moi une révélation fantastique, d'un appareil qui réunissait en un seul appareil, un clavier, un écran de télévision et un lecteur de bande magnétique, deux langages de programmation (BASIC et APL), le tout, particulièrement clairement documenté. Si, en l'emmenant pour le tester, je constatais que sa « portabilité » demandait quelques efforts physiques, le lundi matin, en présentant à mes collègues le résultats d'autres efforts, j'étais absolument conquis par ce système et surtout par le langage APL qui m'avait permis d'obtenir, peut-être

pas d'une façon très académique, un résultat qui aurait demandé un délai de réalisation plus important et la collaboration de plusieurs personnes par les moyens habituels.

A la fin de cette année 1978, j'ai pu convaincre ma direction de l'adaptation de cet équipement à nous aider et obtenir d'elle l'autorisation de se le procurer. Par contre, pour satisfaire aux règles de bonne gestion auxquelles nous nous soumettions, j'ai dû trouver et acquérir un matériel d'occasion, et accepter, en plus, de prendre en charge l'informatisation du traitement d'une paye.

Ces opérations ont démarré en 1979. Il est vite apparu nécessaire d'adjoindre à l'unité centrale un dérouleur de bande, et, ensuite, une unité auxiliaire de deux minidisques.

Ces équipements ont été remplacés successivement par un modèle 5110 puis par un PC AT avec le recours à un convertisseur annexe de disquettes. A cette dernière occasion, j'ai migré de l'APL IBM à l'APL STSC.

J'ai ainsi programmé jusqu'à la fin des années 80 les deux principales applications suivantes

La paye :

Les paies de toutes les catégories de salariés de l'entreprise étaient traitées manuellement au moyen d'un système classique par décalque. Pour des raisons de confidentialité, la direction voulait isoler le traitement des paies d'une catégorie de personnel (Environ 200 personnes) et recherchait à réduire au maximum la charge que cela représentait.

L'objectif principal de cette application a été de mettre à la disposition des personnes qui en étaient chargées, un ensemble de programmes permettant:

- de calculer mensuellement la paie.
- d'introduire dans le système, des paies pré-calculées manuellement.
- d'éditer les journaux de paie par catégorie.
- d'obtenir les statistiques mensuelles, afin notamment de calculer les charges sociales.
- de préparer les centralisations comptables mensuelles.
- de préparer les déclarations annuelles.

Les programmes de cette application étaient regroupés dans les chapitres suivants :

- Fichier central du personnel
- Eléments fixes employeurs
- Eléments fixes salariés
- Elément variables salariés
- Calcul automatique de la paie

- Journaux de paie provisoires ou définitifs
- Paies pré-calculées manuellement
- Journaux de paie définitifs
- Bulletins de salaires
- Statistiques mensuelles et annuelles.
- Centralisation comptable

Cette application a fonctionné de janvier 1980 à décembre 1989 et la seule grosse modification que j'ai dû lui apporter a été, en décembre 1988, de faire figurer sur les bulletins de salaires, la part de charges sociales des employeurs et la situation des congés payés et repos compensateurs.

Le budget:

Il s'agissait d'établir les comptes de résultat pour environ 200 présentations différentes, appartenant à une quarantaine de produits, analysables selon les critères suivants:

- Départements: produits d'entretien, pharmacie, cosmétiques,...
- Monnaies: francs français et dollars
- Types d'exercices: réel, budget, «forecast», prévisionnel actualisé,..
- Types de données: internes et externes.

Pour les présentations, les données et les résultats étaient analysés dans des matrices qui comportaient

- 14 colonnes :
 - 1 pour la référence du produit auquel appartient la présentation.
 - 1 pour la référence de la présentation
 - 1 pour chacun des mois de l'exercice
- Autant de lignes que de présentations.

Il y avait une matrice pour les quantités vendues, produites et chacun des postes du compte de résultat, du chiffre d'affaires brut à la marge brute, en analysant les composants du prix de revient: matières premières, temps unitaires de production, main-d'œuvre directe et frais généraux de production.

Pour les produits, les données et les résultats étaient regroupés automatiquement à partir des chiffres par présentation, dans des matrices qui comportaient:

- 13 colonnes: 1 pour la référence du produit

1 pour chacun des mois de l'exercice
- Autant de lignes que de produits.

Il y avait autant de matrices que pour les présentations (exceptées celles concernant les quantités, les prix et temps unitaires) avec en plus, celles nécessaires pour chacun des différents médias de publicité.

Ces matrices permettaient d'éditer des états comparatifs de données pour des périodes mensuelles et cumulées, en analysant les écarts pour déterminer ce qui relevait des variations de volume et ce qui était provoqué par des différences de prix unitaire.

Autour de cette application se sont greffés différents programmes nécessaires à l'approfondissement des prévisions comme

- un programme de comptabilité industrielle pour la détermination des frais généraux de production et des taux horaires par sections des unités de production.
- un programme de gestion des participations publicitaires avec la grande distribution.
- un programme de gestion de trésorerie.

Au cours de ces années, la faculté de disposer en permanence de ce langage de programmation, m'a toujours permis de pouvoir modifier aisément ces applications, pour les adapter aux circonstances, rapidement, avec un degré de sécurité appréciable et de faire preuve d'adaptabilité.

Après avoir cessé pendant une dizaine d'années toute activité de programmation, je recommence aujourd'hui à le faire pour de nouvelles applications (gestion de frais de déplacement, opérations boursières,...). Bien que les tableurs actuels résolvent correctement une partie des travaux évoqués ci-dessus, je me sens beaucoup mieux équipé avec, dans ma boîte à outils, l'outil APL dont j'ai acquis récemment, une version nouvelle, compatible avec Windows 95. Je ne manquerai pas de vous relater les péripéties de cette nouvelle expérience.

Les concepts généraux des I*Net

par Bernard Mailhol

1 PRESENTATION

Internet et le Web sont devenus des outils que la planète entière utilise.

Cet intérêt semble soudain, un phénomène de mode, qui passera.

Je pense qu'il n'en est rien, que le Web répond à des besoins anciens, formulés, et constitue une réponse simple et générale à ces besoins.

Toutefois, ces concepts sont entourés d'un grand mystère, sont enrobés dans de grands mots. Tout informaticien d'aujourd'hui, quelle que soit sa spécialité, doit connaître ces concepts, et les utilisera bientôt.

Ce texte reprend des présentations faites aux journées Ipso des 14 et 15 Octobre 1997, et à la journée AFAPL du 6 novembre 1997. Il a été créé à partir de ces présentations, par ajout de commentaires aux images proposées dans le support de l'exposé.

Ce support est aussi disponible en <http://www.mailhol.com/pub/ipso> sous forme de pages html.

Vous y trouverez aussi les exemples, que vous pourrez appeler depuis votre fureteur.

2 I*NET

L'ensemble de ces réseaux repose sur le protocole de transmission TCP/IP. Ce protocole, selon son contexte d'emploi, peut prendre différentes formes, avoir différentes propriétés.

2.1 Internet Intranet, Extranet

Beaucoup de mots, anciens ou nouveaux sont utilisés. Ils correspondent en fait à des concepts nouveaux, qu'il faut bien nommer.

1 - *Internet* LE Réseau planétaire

Ce réseau n'est pas récent. Il s'est développé dans le monde universitaire depuis plus de 20 ans. Ce qui est nouveau, c'est son ouverture sur l'ensemble de la population.

Alors que les universitaires étaient les principaux utilisateurs de ce réseau, il s'ouvre maintenant aux entreprises et aux particuliers. Ce réseau devient un réseau marchand.

2 - *Intranet* Réseau d'entreprise

Un Intranet est un réseau interne à l'entreprise (ce qui n'est pas nouveau), mais dont l'usage est semblable à celui de l'Internet planétaire.

Mêmes outils, mêmes habitudes. La différence vient de la confidentialité apparente du réseau, ce qui permet d'y placer des données de l'entreprise, plus sensibles.

3 - *Extranet* Réseau fédéré d'entreprises

Internet étant fondé sur TCP/IP, ce réseau ne dispose pas de la notion de "qualité de service". Aucun usage de ce réseau ne peut garantir ni confidentialité, ni débit minimum.

Un extranet est un réseau reliant plusieurs entreprises, mais distinct d'Internet, de façon à assurer une confidentialité, et garantir un débit.

Un extranet est géré par les mêmes outils que l'Internet, ou un Intranet.

Repose sur TCP/IP, Protocole de transmission utilisé en Internet, *très largement* utilisé dans les Intranets (On peut utiliser un Intranet via un protocole autre que TCP/IP, mais cette particularité se rencontre assez peu).

L'usage d'un tel réseau fait appel à quelques notions, qui n'ont pas à rester mystérieuses (toujours un problème de pouvoir !)

- L'accès à une machine (*host*) repose sur l'usage de noms de domaine et les **DNS** : Domain Name Servers.

Vous utilisez un réseau TCP/IP depuis une machine personnelle, une station de travail, ou depuis un serveur regroupant des centaines d'utilisateurs. Dans tous les cas, cette machine s'appelle (en anglais) *uHost*.

Chaque *Host* est repéré par une *adresse IP* (comme tout téléphone est repéré par son numéro). Si un numéro de téléphone est, en France, de 10 chiffres, une adresse IP est composée de quatre nombres compris entre 0 et 255, séparés par un point. L'adresse 10.21.56.35 est une adresse IP.

Il n'est pas facile de se souvenir d'une adresse IP numérique. Chaque host peut porter un nom, aussi appelé *nom de domaine*. Il existe alors des serveurs chargés de transformer un nom de domaine en adresse IP.

Ces programmes s'appellent de **Domain Name Servers (DNS)**

Lorsque vous installez TCP/IP sur votre machine, vous devez indiquer quelle est son adresse IP, et quelle est l'adresse IP du DNS le plus proche, pour pouvoir retrouver toutes les autres adresses IP du réseau concerné.

- TCP/IP est régi "collégalement", et décrit par de **RFC : Request For Comments**

L'une des particularités de TCP/IP est d'être un produit non commercial. TCP/IP est défini et étendu collégalement, au-delà des intérêts financiers et des démarches stratégiques de tel constructeur.

Le processus d'évolution est régi par des publications, acceptées par cette association d'utilisateurs. Chaque publication s'appelle un(une ?) *RFC - Request for Comments* et dispose d'un statut : proposition, facultatif, obligatoire ... Aujourd'hui, plus de 2000 RFC existent, toutes consultables sur Internet.

2.2 Sécurité et confidentialité (Internet)

La grande rusticité de TCP/IP - ce qui a participé à sa diffusion - présente certains inconvénients, vis à vis des besoins de l'entreprise.

En pratique, les manuels traitant de la sécurité montrent tous que :

- 1 - Il est *raisonnable* de penser qu'un ou plusieurs systèmes analysent toute transaction.
- 2 - Il est *raisonnable* de penser que 20 % des messages échangés sont stockés quelque part.
- 3 - *Le plus souvent*, le nom de notre interlocuteur est celui qu'il annonce. (Mais de nombreuses exceptions)

Ces remarques montrent que l'usage d'Internet doit être mesuré, les risques analysés. Les risques sont d'autant plus grands que les données sont sensibles pour l'entreprise.

De fait, une transmission TCP/IP se fait :

- 1 - Sans confidentialité
- 2 - Avec des risques de transformation du message
- 3 - Avec un interlocuteur inconnu

L'importance de ces problèmes doit être prise en compte dès le début

2.3 Sécurité et confidentialité (Intranet)

On peut penser a-priori que ce genre de problèmes ne se rencontre pas en Intranet. Ceci est souvent vrai, mais ...

Si des personnes sont curieuses, malveillantes, ou de mauvaise humeur (au moins pendant quelques minutes)

- 1 - Les règles ci-dessus sont à prendre en compte
- 2 - Mais les données sont beaucoup plus critiques.

L'importance de ces problèmes doit être prise en compte dès le début

Dans le cadre de cet article, je ne peux me permettre de développer plus avant ces problèmes. Des exemples montrent qu'ils sont réels.

2.4 Travaux sur la sécurité

Cette faiblesse des protocoles de TCP/IP a, en contrepartie, provoqué le développement rapide et public de nouveaux domaines de l'informatique, concernant la sécurité.

Ces travaux reposent tous sur l'hypothèse de transmissions non confidentielles, éventuellement transformées, avec un tiers inconnu (du moins le système de transmission ne peut nous donner son identité, avec certitude).

Les questions auxquelles ces travaux se proposent de répondre sont :

- Comment assurer l'identification et l'authentification des interlocuteurs quand on ne peut croire le réseau ?
- Comment s'assurer que le message reçu est bien le message envoyé ?
- Comment protéger la confidentialité de nos échanges, quand ils sont surveillés, et éventuellement transformés ?
- Comment assurer la trace d'une transaction, quand elle est importante ?

Les systèmes retenus peuvent alors répondre à une ou plusieurs de ces questions, selon le coût accepté, et la législation en vigueur dans le pays de leur usage.

En particulier, la protection placée peut le plus souvent être cassée par un travail plus ou moins conséquent. Il faut alors déterminer le niveau de protection à appliquer :

- Vis à vis de la mauvaise humeur d'une personne
- Vis à vis d'une journée d'effort, en 1997
- Vis à vis d'une semaine d'effort, en 1997
- Vis à vis d'un organisme intéressé
- Vis à vis d'un état.

Chacun sait que les moyens d'investigation sont de plus en plus puissants. Une protection d'une semaine aujourd'hui ne sera plus qu'une protection de quelques heures demain.

3 L'USAGE DE CES RESEAUX

Avant d'évoquer les applications d'Internet, quelques mots à propos du réseau lui-même.

3.1 Applications standard en TCP/IP

La mise en place d'un TCP/IP sur une machine installe non seulement les programmes de transmissions, mais aussi des applications standard (définies évidemment par des RFC).

Les applications les plus fréquemment utilisées sont aujourd'hui

- 1 - Telnet (émulation de terminaux, de 3270)
- 2 - Ftp (transfert de fichiers)
- 3 - Sntp, pop3 (courrier)
- 4 - Rpc, nfs ... (sur des réseaux locaux essentiellement)
- 5 - Sntp gestion des incidents, surveillance du réseau
- 6 - Web -->> application émergente

Vous pouvez aussi développer vos applications à base d'une programmation "en sockets".

3.2 Les origines du Web

Depuis de nombreuses années, la recherche de l'accès universel à la connaissance fascine de nombreux chercheurs.

La notion d'*hypertexte*, permettant de parcourir un texte dans un ordre non linéaire, au choix du lecteur et selon ses connaissances antérieures, ses centres d'intérêt, date de plus de 30 ans (le terme Hypertexte a été créé par Théodore Nelson, en 1967, Englebart, 1963 ...)

Des programmes ont été développés, sur des machines isolées (Hypercard sur Mac, par exemple), pour mettre en place cette notion.

Dans le monde Internet, plusieurs produits ont eux aussi permis des accès généraux à des bases de connaissance : GOPHER [créé par l'Université du Minnesota] permet ces interrogations, mais sous une forme peu agréable.

Le concept du Web, et de ses fichiers html, a été créé par le CERN à Genève, vers la fin des années 1980, pour donner lieu à des produits dès le début des années 1990. Mis en place dans le réseau Internet, ce produit pouvait présenter le réseau de l'époque comme un gigantesque réservoir de savoir universitaire.

Ce concept du Web répond à des besoins exprimés des entreprises, justifiant à lui seul la mise en place d'un *Intranet*.

1 - Besoin de fédérer plusieurs sources d'information

La tendance dans les entreprises est de multiplier le nombre de serveurs, pour ne retenir que des serveurs spécialisés, mono-applications.

En face de cette inflation de machines, les utilisateurs doivent conserver le pouvoir d'accès aux données.

Le concept de la navigation du Web permet ainsi de mettre en place des accès généraux aux données.

2 - Besoin de disposer d'un interface très élémentaire

Il est aussi souhaitable que l'usage de ces outils généraux d'accès aux données dans l'entreprise soit le plus élémentaire possible : chacun doit maintenant disposer d'un choix de données de plus en plus vaste. L'interface doit alors pouvoir être la plus sommaire possible.

une action principale : cliquer.

En effet, le plus souvent, il suffit de désigner (par le pointeur) l'information souhaitée, et cliquer pour obtenir cette information. Les écrans tactiles simulent un clic de souris, et sont alors directement utilisables dans le Web.

3 - Besoin de disposer d'un outil standard.

Enfin, la préparation de ces données doit être standardisée, être régie par des normes, pour assurer une pérennité de l'application sur plusieurs années.

Les fichiers utilisés par le Web sont des fichiers balisés (dont la description est conforme à la norme ISO : SGML), ne contenant que des caractères imprimables, indépendants (ou presque) de tout constructeur.

Ces fichiers peuvent être créés de différentes façons, y compris par des outils et des procédures développées par les entreprises elles-mêmes. De tels fichiers balisés indiquent la *signification* des données, et non leur *présentation*. On peut donc reprendre un fichier balisé pour le transformer dans un autre système, car on connaît la sémantique du contenu.

Ces fichiers balisés ne sont pas récents, et certains les pratiquent depuis plus de 20 ans.

Le choix des balises présentes dans un fichier html permet de préparer un affichage plaisant, et offrant aussi des saisies limitées.

3.3 URL - Uniform Resource Locator

La notion de base apparente à l'utilisateur est la notion d'**URL**, désignant toute ressource accessible par le Web.

Cet **URL** est constitué de trois parties:

`http://www.xyz.com:80/ipso/v97_concepts/index.html`

1 - Le protocole

http://

(on peut aussi choisir *mailto:ftp:...*)

2 - L'adresse de son serveur (nom de domaine)

www.xyz.com:80

(éventuellement le numéro de port, si ce dernier est différent de 80)

3 - L'identification de la ressource vis à vis du serveur désigné.

/ipso/v97_concepts/index.html

(Les serveurs contiennent des tables autorisant ou refusant l'accès à ces ressources)

L'essentiel d'une session de consultation en Web est la fourniture initiale d'un URL, l'examen du contenu de ses informations, puis l'appel d'une autre ressource.

La mise en place d'un tel système repose sur deux conventions essentielles

1 - Le protocole de dialogue entre client et serveur (http)

2 - Le format des fichiers transférés (html)

4 HTTP - HYPERTEXT TRANSFER PROTOCOL

HTTP désigne les règles que doivent suivre client (fureteur) et serveur pour dialoguer : c'est un *protocole*.

4.1 Protocole et serveur

Ce protocole décrit les messages échangés entre le *fureteur* et le *serveur* (httpd)

Ce protocole, dans sa version 1.0, est fondé sur l'usage de courtes (et nombreuses) sessions TCP/IP.

1 - Le client demande la transmission de données, précisément identifiées par un URL.

Le message reçu par le serveur est assez court. Il ne contient que des textes directement imprimables.

2 - Le serveur prépare et transmet une réponse

C'est un fichier de texte, une image, un son ...

3 - Le serveur clôt la session.

Si le fichier transmis est une page Web, cette page peut contenir le chargement d'autres ressources, comme des images par exemple. La transmission de chaque image d'une page Web provoque alors une nouvelle session TCP/IP, de courte durée.

4.2 Usage des URL en serveur

Le serveur peut rechercher un fichier à partir de l'URL :

1 - Déclaration des ressources reconnues

La configuration du serveur contient des directives permettant d'associer un sous-répertoire au début de la définition de la ressource demandée. (ces exemples sont donnés dans le cadre d'un système de fichiers semblable à celui d'Unix, ou de DOS).

```
Pass /ipso/v97_concepts/*      x:\web\ipso\vanves97\concepts\*
```

2 - Transformation d'une requête en nom de fichier

A partir de ces déclarations, une demande de ressource peut être transformée en nom de fichier, par le serveur.

http ://www.xyz.com:80/ipso/v97_concepts/index.html

... transformé en

x:\web\ipso\vanves97\concepts\index.html

Par le serveur sur la machine dont le nom de domaine est : ***www.xyz.com*** à l'écoute sur le port **80**.

4.3 Message reçu par le serveur

Le serveur reçoit ce message, terminé par une ligne vide

```
GET /ipso/v97_concepts/index.html HTTP/1.0 (new line)
Accept: text/html (new line)
Accept: image/gif (new line)
... (quelques dizaines de lignes)
User-Agent: <programme et version du fureteur> (new
line)
(new line)
```

Les premiers caractères reçus sont "GET(espace)", transmis en code ascii. Ce code identifie le sous-protocole de http retenu pour cette demande de transmission de ressource.

Les caractères suivants reprennent l'identification de la ressource, extraite de l'URL

Après un espace, le fureteur envoie la version du protocole HTTP qu'il sait analyser, puis termine sa première ligne par un caractère de "retour chariot".

Les caractères qui suivent ce caractère de "retour chariot" forment une autre ligne de texte. Plusieurs lignes décrivent ainsi les capacités du fureteur, son nom. Le serveur peut utiliser ces informations pour construire sa réponse. Ces définitions des capacités du fureteur sont jointes à chaque demande de ressource, le serveur ne dispose d'aucune mémoire des accès précédents.

La fin de la demande est marqué par une ligne vide (un second caractère de retour chariot).

4.4 Réponse du serveur

Le serveur retourne ces données, et clôt la session

```
HTTP/1.0 200 Document follows (new line)
Server: XXXX
Content-type: text/html(new line)
Content-length: 224(new line)
(new line)
<Contenu de la réponse, ici un flot de          >
<données en format html, de 224 octets de long >
```

Le serveur renvoie un message, lui aussi composé de caractères ascii imprimables. Le début de la réponse reprend le même schéma de lignes de textes, terminées par un caractère de "retour chariot". La fin de l'en-tête de sa réponse est marquée de la même façon par une ligne vide (deux "retour chariots" consécutifs).

- La première ligne donne le code retour : 200 est le code d'acceptation.
- Les lignes suivantes décrivent le serveur, le format des données transférées.

Ces définitions suivent les règles de *header Mime*.

Un serveur complet dispose de nombreuses autres fonctionnalités, de protection des ressources, de transformations des réponses, de cache d'autres serveurs ...

Un serveur élémentaire peut se contenter de cette fonctionnalité. Vous pouvez développer une application qui agit comme un serveur limité, fournissant des données d'entreprise spécifiques (voir en <http://www.mailhol.com/pub/afapl> un exemple de développement en APL d'un tel serveur).

4.5 HTTP 1.0 et HTTP 1.1

HTTP est officiellement défini, en V 1.0 par la RFC 1945 (août 1996)

Ce protocole a l'avantage de sa rusticité. Toutefois, il présente certains inconvénients.

1 - Nombreuses sessions (lourde charge des routeurs et DNS)

Chaque demande de ressource fait appel à une nouvelle session TCP/IP. Par exemple, une page contenant 10 images peut créer 11 sessions TCP/IP, en l'espace de quelques secondes.

Ces ouvertures de session alourdissent inutilement le réseau, les routeurs etc..

2 - Les fureteurs ne peuvent savoir si une demande identique produira les mêmes effets.

Certains documents rendus ne sont mis à jour qu'exceptionnellement, d'autres le sont chaque minute. Le fureteur ne sait pas différencier ces documents, et ne peut facilement gérer la mémorisation locale des ressources, dans *son* cache.

Un serveur intermédiaire (*proxy*) rencontre les mêmes problèmes, et ne connaît pas précisément la durée de vie de chaque ressource copiée dans les fichiers locaux.

3 - Protection (mot de passe) très faible.

La protection des accès par mots de passe pourrait être utile. Sa mise en oeuvre en HTTP fait que ce mot de passe est envoyé par le fureteur - très légèrement codé - lors de chaque demande de ressource. L'appel d'une page et de ses 10 images provoque la transmission du mot de passe ... 11 fois.

Ces points sont pris en compte dans HTTP 1.1 (à l'état de proposition) décrit par la RFC 2068 (août 1996)

... et déjà utilisée.

Le système de protection par mots de passe est décrit en RFC 2069, mais reste sommaire : "[this system] is better than nothing" (ligne 157 de la RFC 2069)

5 HTML - HYPERTEXT MARKUP LANGUAGE

HTML décrit le format des données transmises du serveur vers le fureteur.

5.1 Un langage de balises

Chaque "fichier" reçu contient la description d'une page affichable.

Il contient un texte "en clair" ne contenant que des caractères imprimables, décrivant la **présentation** et le **comportement** de la page.

Le fait que ce texte ne contienne que des caractères imprimables fait...

- Qu'il ne peut rien cacher, la conformité aux normes d'une page html est facilement détectée.
- Qu'il peut facilement être créé par une programmation simple.

Une application spécifique de faible coût peut préparer des pages, systématiquement. A titre d'exemple, on peut préparer les principaux éléments d'un manuel de référence d'une application à partir des tables pilotant l'application elle-même. La préparation de la documentation d'une application écrite en *Java* en est une illustration. Certains produits d'aide au développement préparent une documentation au moyen de fichiers balisés, générés automatiquement.

- Qu'il peut être maintenu, pendant plusieurs années.

La définition des balises est telle que - en se limitant à l'usage de la norme, évidemment - une page Web sera encore affichable dans quelques années. Son aspect pourra être un peu désuet, mais le contenu sera valide.

Cette pérennité est une propriété indispensable à la mise en place d'outils de documentation et d'information devant être opérationnels à l'échelle de temps des entreprises.

La forme actuellement applicable est décrite par le **W3 Consortium** et en est à la version 3.2. Les pages du Web Ipso (<http://www.ipso.org>), quand elles arrivent dans le fureteur, suivent rigoureusement cette norme.

5.2 Structuration d'une page

Le but de cet article n'est pas d'entraîner à la préparation de pages Web. Il n'est pas de donner une définition complète de son format. Il est seulement de montrer les principaux éléments de ce langage html.

Tout d'abord, un document Web est composé de deux parties :

- Une en-tête, paramétrant le document
- Le corps du document.

Les *balises* sont indiquées par des signes < et >. Une balise de fin est indiquée par un / ex : <title> et </title>

Voici un document sommaire, composé d'un titre et de deux paragraphes.

```
<html>

<head>
<title>Première page</title>
</head>

<body>
<h1>Structure d'une page</h1>
<hr>
<p>Un paragraphe
<p>Un second paragraphe
</body>

</html>
```

La ligne contenant le titre contient un "è". Ce caractère spécial, représenté par un code ISO latin-1 supérieur à 128 peut poser problème en cas de transmission de la page en FTP, en mode ascii. Il est préférable de le remplacer par un symbole représentant ce caractère :

```
<title>Premiè&egrave;re page</title>
```

Vous trouvez cet exemple sommaire en

http://www.mailhol.com/pub/ipso/v97_concepts/demostr.html

5.3 Appel d'une image

Html permet l'appel d'images dans une page. Cette image peut avoir un but informatif; elle peut aussi - c'est très important - avoir un but esthétique.

Dans l'exemple ci-dessous, une image (ipsofond.gif) est placée en fond d'écran; une seconde image est affichée. Il s'agit d'un *gif* animé qui, en un seul fichier, appelle plusieurs images, en séquence.

Si les images rendent une page Web plus agréable à regarder, elles peuvent aussi provoquer des temps de transmission importants, et la lassitude de l'utilisateur. Faites attention au volume des données transmises.

```
<html><head><title>Seconde page</title></head>
<body background='ipsofond.gif'>
<h1>Appel d'une image</h1>
<hr>
<p>Un paragraphe

<img src='image.gif'>

<p>Un second paragraphe
</body> </html>
```

Vos trouvez un exemple en

http://www.mailhol.com/pub/ipso/v97_concepts/demoimg.html

Cet exemple place aussi deux points bleus devant les items "Premier paragraphe", et "Second paragraphe".

Cet exemple utilise des images issues du même serveur. Une image est en fait une ressource, définie par un URL. Rien n'empêche d'aller chercher une image sur un autre site. Il suffit de préciser son URL complet.

5.4 Appel d'un lien

L'un des aspects fondamentaux de html est de pouvoir appeler d'autres pages html. Toute page html peut ainsi appeler d'autres pages, situées sur le même serveur, ou sur tout autre serveur.

Cet appel de lien se fait en trois phases :

1 - ``

Cette commande donne l'ancrage du lien. En paramètre, on précise l'URL de la page qui doit être appelée quand ce lien est activé.

2 - Texte

Ce texte est affiché d'une couleur différente. Le plus souvent, lorsque le pointeur passe sur ce texte, il change de forme, pour devenir une main dont l'index est tendu !

Au clic, la page dont l'URL est indiqué est demandée au serveur ad'hoc.

Ce texte peut aussi être remplacé par une image.

3 - ``

Cette balise vient indiquer la fin de la zone active.

```
<html><head><title>Troisième</title></head><body>
<h1>Essai de link</h1>
<hr>
<p>Un paragraphe

<p>Un second <a href="demostr.html">paragraphe</a>

</body></html>
```

Vos trouvez un exemple en

http://www.mailhol.com/pub/ipso/v97_concepts/demolnk.html

5.5 Appel d'un lien sur image

On peut aussi afficher une image et découper cette image en plusieurs zones, de telle sorte que la ressource appelée soit dépendante de la position du pointeur dans l'image.

Le texte suivant montre le découpage de l'image en plusieurs polygones.

```
<html><head><title>Exemple HTML</title>
</head>
<body bgcolor=yellow link=red vlink=red>
<h1>Essai de lien sur image</h1>
<hr>


<!-- Description de la topologie de l'image -->
<map name="voiles">
<area shape=poly coords="74,0,76,61,65,179,131,209,135,0"
href="Le_spinaker" alt="Le Spinaker">
<area shape=poly coords="0,0,0,200,100,200,100,0"
href="Le_ciel" alt="Le ciel">
<area shape=poly
coords="11,319,61,264,111,232,136,231,212,262,212,319"
href="La_coque" alt="La coque">
<area shape=poly coords="137,0,127,218,212,219,212,0"
href="La_grand_voile" alt="La Grand voile">
<area shape=default nohref>
</map>

</body></html>
```

Vos trouverez un exemple en

http://www.mailhol.com/pub/ipso/v97_concepts/demomap.html

Cet exemple reprend le texte ci-dessus, mais prépare un tableau de 2 colonnes. Il place un texte explicatif dans la première colonne, puis l'image dans la seconde.

Les principaux éléments de cette image (coque, spi, grand voile, ciel) correspondent chacun à un appel différent. Le reste de l'image (la mer) ne correspond à aucun lien. Lorsque le pointeur la désigne, il reprend sa forme habituelle.

L'exemple affiché vous permettra aussi de tester la conformité de votre fureteur avec la définition de HTML 3.2, car les ordres utilisés n'existaient pas en html 2.0.

5.6 Saisie en formulaire

Enfin, html permet des saisies sommaires. Ces saisies peuvent permettre de préparer des applications transactionnelles, ou simplement de permettre des choix plus rapides qu'une suite de clics sur des pages qu'il faut appeler en cascade.

Une saisie se fait entre les balises `<form>` et `</form>`

Entre ces deux balises, des textes affichés, et des balises `<input ...>` décrivent des champs en entrée ou des boutons divers.

Le résultat des saisies est transmis vers le serveur, la transmission comprenant l'ensemble des valeurs saisies. Le traitement est au choix du serveur.

```
<html><head><title>Troisième</title></head><body>
<h1>Essai de formulaire</h1>
<hr>
<p>Début de saisie

<form method=get action="/cgi-bin/ipso/cgiget">
<p>Machine : <input type="text" name="machine" size=20>
<p>Localisation: <input type="text" name="location" size=20>
<p>Cliquez pour finir
<input type="submit" value="Envoi">
<input type="reset" value="Annuler"
</form>

</body></html>
```

Vos trouvez un exemple en

http://www.mailhol.com/pub/ipso/v97_concepts/demoinp.html

Lorsque vous demandez l'envoi de vos saisies, le script CGI indiqué est absent. Le but est de montrer l'appel, non d'exécuter la requête. (voir ***http://www.mailhol.com/pub/ipso/v97_cgi*** une présentation des script CGI, et un appel effectif).

Attention, la page html peut être modifiée par l'utilisateur. Les données effectivement transmises au serveur peuvent être différentes des données attendues. La procédure appelée doit être largement testée et insensibilisée aux transmissions de paramètres inattendus.

5.7 Traitements dynamiques

Nous n'avons présenté que la possibilité d'aller chercher des pages préparées en fichier, sans aucune possibilité de traitement.

Des traitements sont cependant possibles

1 - Dans le serveur

*Un programme peut prendre en compte tout ou partie de la requête : script CGI

Ces scripts CGI sont présentés en http://www.mailhol.com/pub/ipso/v97_cgi.
Ils permettent l'appel de programmes en réponse aux actions des utilisateurs.

2 - Dans le fureteur

*Options de saisie (html 3.2)

Les options de saisie permettent des saisies sommaires, transmises vers le serveur, à destination d'un script CGI.

*Exécuter des programmes

Le fureteur peut exécuter des programmes. Des questions immédiates se posent :

1 - D'où viennent ces programmes ?

2 - Quels sont leurs droits ?

3 - Que font ils ?

4 - Peuvent-ils me nuire ?

Au moindre danger, on peut citer

•Les applets de Java (html 3.2)

La notion de sécurité fait partie intégrante de la définition de Java (marque déposée de Sun). Les concepts de base supposent même que la machine sur laquelle s'exécutent les programmes Java (les *applets*) ne dispose pas d'un hardware permettant de protéger les instructions privilégiées !

Attention, il semble que de la désinformation circule sur le Web à ce propos. (se reporter à la littérature écrite, signée, authentifiée).

•Un script Javascript (rien à voir avec Java, et hors norme html 3.2)

Il semble que les scripts Javascript ne disposent pas de grands pouvoirs de destruction.

Le fait que Javascript soit complètement interprété implique que le programme est transmis en clair, donc contrôlable en cas de doute. En cas de problème avéré, on peut alors se retourner judiciairement vers le "directeur de publication", déclaré en préfecture lors de la création du site Web., si ce site est un site français.

6 INSTALLONS

Après cette présentation générale, quelle procédure faut-il suivre pour installer un Intranet ?

6.1 Les outils nécessaires

L'équipement et les connaissances nécessaires à la mise en place d'un Web sont limitées.

1 - Un réseau TCP/IP

Ce réseau peut être construit sur plusieurs supports. Parmi ceux-ci, on peut citer :

*LAN (tous supports)

*WAN X25

*RTC / numéris + protocole de transport PPP (le plus souvent)

2 - Un (plusieurs) serveurs "httpd"

Chaque serveur "httpd" est une machine, dont la puissance dépend de la charge qu'il devra traiter. Cette machine peut être un PC pour un serveur de faible usage, ou bien un site central de grande puissance, pour traiter des milliers d'appels simultanés.

3 - Chaque poste installé (le plus souvent un PC, ou un Mac)

*Dispose de TCP/IP

*Dispose d'un fureteur Web.

4 - Savoir administrer le réseau et le serveur

Les compétences sont ici nécessaires, et dépendent de la taille du réseau. Un réseau ne comprenant qu'un seul réseau local de quelques postes ne pose pas trop de problèmes, mais la connexion de plusieurs réseaux, l'évolution harmonieuse de l'ensemble demande des compétences réelles.

5 - Savoir créer et mettre à jour les pages Web. Enfin, il faut savoir créer des pages Web.

Certains choix sont importants :

- a - Comment structurer les accès au serveur ?
- b - Comment créer initialement les pages
- c - Comment assurer la maintenance et l'évolution du site ?

6.2 Quelques repères

Adresses intéressantes

1 - **Internic** <http://ds0.internic.net/>

Page d'accueil d'Internic

*Registration services

*Information & éducation services

*Directory & database services

*Net Scout services

Liste des RFC <http://ds0.internic.net/rfc/>

(2200 textes ou flots postscript)

2 - **Description de HTML**

Page d'accueil <http://www.w3.org>

Propose la description de la prochaine version (4.0 en juillet 97 ... mais ce n'est qu'une proposition)

Référence HTML 3.2

**<http://www.w3.org/pub/WWW/TR/REC-html32.html>*

**<http://www.w3.org/pub/WWW/MarkUp/Wilbur>*

**<http://www.w3.org/pub/WWW>*

Description de http:

<http://www.w3.org/pub/WWW/Protocols/>

3 - Problèmes de sécurité, vulnérabilité

CERT Coordination Center
Software Engineering Institute
Carnegie Mellon University
Pittsburgh, USA.

<http://www.cert.org>

4 - Analyse de serveurs Web *<http://www.Webcompare.com>*

Donne (aussi) une liste de serveurs

Attention, contient beaucoup de "Pub",

5 - Serveurs d'IBM:

<http://www.ics.raleigh.ibm.com/>

6.3 Vocabulaire

Attention, le vocabulaire s'étend.

•Spoof (dans la RFC 2068)

Ce mot, rencontré dans la RFC 2068 (description de HTTP 1.1) est nouveau en informatique. Ce mot anglais signifie *arnaque*. On cite souvent "*l'arnaque au DNS*", dans laquelle le serveur de nom donne de fausses informations.

•Parano

Les livres traitant de sécurité assimilent les personnes chargées de ce problème de paranoïaques.

Cette qualification est surprenante, car d'une part elle confond une personne et sa fonction, et d'autre part, elle nie au passage tout problème de sécurité, tout en montrant par des exemples les failles innombrables des systèmes, en citant des cas de nuisance.

APL98 à Rome

par Paolo Di Chio

Ndlr: Le congrès APL98 aura lieu à Rome du 27 au 31 juillet 1998.

Il est organisé par SIGAPL italien, voilà l'information qu'ils ont envoyé sur mon e-mail :

APL98 Rome

July, 27th-31st, 1998

University of Rome "Tor Vergata" - Faculty of Economics

The Array Processing Languages Conference

"New gems from old roots"

Appel aux communications

Les centres d'intérêt de ce congrès sont suivants (sans être limités à ces sujets)

1. State of the art of APLs:
Present situation and future directions
2. Computer Science:
Human Computer Interaction, Object-Oriented Programming,
Parallel Architectures, Parallelism and Concurrency,
Distribution,
Meta-level Programming, INTERNET Computing
3. Discrete Mathematics, Algorithms and new computing paradigms
(Neural Networks, Genetic Algorithms, DNA computing)
4. Applications:
Finance and Financial Maths
Economics and Social Sciences
Insurance and Actuarial Maths
Statistics and Operational Research
Image Processing
Simulation of physical, biological and social phenomena
Others
5. Education: Teaching APLs, Teaching with APLs

Les types de contributions sont suivants:

1. Papiers 30/45 minutes
2. Formations 60/90 minutes
3. Ateliers 90 minutes
4. Poster/panel sessions
5. Birds-of-a-feather sessions

Pour chaque type les dates limites sont suivantes

Papiers:

January	15th	1 page abstract
February	15th	first acceptance
April	15th	first draft
May	15th	acceptance notification
June	15th	final version

Formations:

March	1st	2 pages abstract
April	15th	acceptance notification
June	15th	final version

Ateliers:

March	1st	2 pages abstract
April	15th	acceptance notification
June	15th	final version

Panel-Posters:

March	1st	1 page description
May	1st	acceptance notification

Birds-of-a-feather:

On site

Soumission:

Abstracts should be submitted on plain ASCII file and sent by e-mail (preferred), fax, or ordinary mail at the following addresses:

E-mail:

mc0307@mclink.it
apl98_abstract@poeco.utovrm.it

Fax:

(+39)-862-432403 (Attn. Prof. Antonio Annibali)

Ordinary mail:

Paolo Di Chio
Via Casetta Mattei 413,
00148 Rome, Italy

Proceedings:

Contributions will be published in the Conference Proceedings.

Pour plus d'information visitez le site web à l'adresse suivante

<http://ibmaix.economia.utovrm.it>

Les joyusetés du correcteur orthographique de « Word »

par La Rédaction

Suggestions du correcteur:

Remplacer «Rudd» par «Rude»

Remplacer «Ralison» par «Raison»

Remplacer «I*Net» par «Internet» (c'est réducteur, n'est ce pas?)

Remplacer « Word » par « Bord » ou « Lord » ou « Nord » (ce qui est certain, c'est qu'il ne perd pas le Nord...)

Remplacer «héliçon» par «hélicon» (et en avant la musique)

Remplacer «pariton» par «parution»

Remplacer «quanton» par «canton»

Remplacer «minouchet» par «manouche» (mais... qu'est ce que c'est?)

Remplacer «Nobels» par «Nobles» (une belle contrepèterie...)

Remplacer «Gabor» par «Gabon» (ce n'est pas la Hongrie!)

Remplacer «ATARI» par «ATTRAIT»

Remplacer «MégaST» par «mégots»

Remplacer «sapiens» par sapins» (sapiens nihil affirmat quod non probet)

Remplacer «translittère» par «transitaire»

Remplacer «taxon» par «talon»

Remplacer «rétrobarre» par «rétrograde» (ah non! surtout pas!)

Remplacer «digitation» par «divination»

Remplacer «récursion» par «réclusion» (ce n'est pourtant pas criminel...)

Remplacer «géniton» par «génitaux»

Remplacer «microcodée» par «microcosme»

Remplacer «forecast» par «forçats» (il insiste!!! est ce dans l'air du temps?)

Remplacer «proxy» par «prose»

Remplacer «title» par «titille»

Remplacer «applets» par «appels»

Remplacer «genus» par «gens» (genus irritabile vatum)