

Java et le Web : Applets

1. Introduction

ous les analystes informatiques se rejoignent pour dire que l'informatique subit actuellement une révolution aussi importante que celle de l'arrivée de la micro-informatique. On en serait à l'aube d'une troisième ère de l'évolution de l'informatique. Après avoir connu la puissance du Mainframe et la liberté du micro-ordinateur, on s'orienterait vers une informatique "orientée réseaux" bouleversant ainsi les équilibres instaurés par les fournisseurs de micro-informatique.

La micro informatique telle qu'on la rencontre actuellement a été pensée comme un outil de productivité individuelle pour compenser les carences des systèmes centraux. Son évolution fut guidée par les principes l'ayant imposée dans l'entreprise: liberté, convivialité, accessibilité, disponibilité. Ce n'est donc que plus tard que l'on s'intéressa à des concepts tels que : l'échange d'informations, la fiabilité et la confidentialité du système. Pour pouvoir exploiter un nouveau Logiciel, il faut en connaître l'origine, la puissance de traitement informatique requise, le type d'ordinateur nécessaire, le format des données,... Cela devient un casse-tête pour les fournisseurs devant maintenir et faire évoluer leurs logiciels sur un grand nombre de plates formes matérielles différentes.

De même pour être capable de récupérer des informations pour les traiter, il est nécessaire de posséder un logiciel pour le texte, un pour les tableaux, un pour les images, un pour les cartes... Ce qui signifie que l'on doit être en possession d'un certain nombre de logiciels et d'avoir un accès à différents types d'ordinateurs suivant la provenance des informations. Il en résulte un nombre impressionnant de formats de stockage et donc des problèmes d'incompatibilités entre les logiciels ne pouvant pas échanger leurs données simplement.

Une solution à ces problèmes est Internet : énorme nébuleuse incontrôlée permettant la connexion universelle et principale responsable d'une révolution logicielle. Si la plupart des utilisateurs s'accordent à dire que les services Internet sont actuellement inadaptés, que les temps de réponse sont parfois assez lents, qu'il y a un manque de fiabilité pour s'attaquer à des transactions financières et que la sécurité n'est pas encore au rendez-vous, tout le monde pense que l'on est au début d'une véritable révolution.

1.1. Internet, WWW, Html?

L'Internet est un réseau reposant sur une architecture distribuée, ne mettant pas en oeuvre de structure centrale ayant un rôle de contrôle du réseau. Ce n'est d'ailleurs pas un réseau, mais un ensemble d'interconnexions de réseaux, ce qui permet d'être souple et évolutif.

Internet utilise le protocole standard TCP/IP : désigne un groupe de protocoles d'Internet, il est en effet suffisamment riche pour permettre à des machines hôtes connectées à différents types de réseaux de communiquer (à condition que ces réseaux soient connectés d'une manière ou d'une autre) TCP/IP permet de dialoguer avec des machines possédant des systèmes d'exploitation différents et des architectures hétérogènes.

Internet sert en fait de vecteur d'accès à des serveurs d'informations distants par des outils d'interrogation navigationnels : c'est le Web. Le Web est basé sur un modèle client/serveur, les objets qu'il manipule sont des fichiers hypertextes de format HTML (Hypertext Markup Language).

Les outils s'appellent les browsers : ils permettent d'échanger des pages d'informations (html) et d'accéder aux serveurs par un protocole standardisé : le protocole http.

1.2. Les Services sont inadaptés !

Ce que recherche en ce moment les utilisateurs, c'est effectivement l'indépendance voire la transparence des plates-formes. Or cette indépendance se retrouve aujourd'hui sur Internet. Seulement Internet, dans son état, est plutôt unidirectionnel. En effet, le Web actuellement est plutôt passif, et ne fournit pas une interaction totale.

Quand le Web deviendra autre chose qu'un outil documentaire, il répondra aux attentes de la plupart des acteurs économiques de l'informatique.

Il permet actuellement de répondre aux besoins de communication, composante essentielle de l'informatique actuelle. De plus, il offre déjà l'indépendance et la transparence vis-à-vis des machines connectées et des systèmes d'exploitation de celles-ci. En effet, personne ne sait et ne se soucie du type de plate-forme d'un utilisateur situé à l'autre bout de l'Internet.

Il ne lui reste plus qu'à devenir réellement applicatif et il répondra aux besoins des utilisateurs. C'est la technologie Java qui est en train de révolutionner le Web et de l'orienter vers un axe applicatif.

2. Java et le Web : Applets

L'intérêt le plus immédiat de Java est de permettre d'intégrer des applications interactives dans des pages Web. Jusqu'à présent, la seule interactivité d'une page Web consistait, en

résumé, à cliquer sur des "liens" soulignés qui renvoient à d'autres pages Web. Grâce à Java, une page Web peut contenir un véritable petit programme, appelé applet[2].

2.1. Définition

Application et une applet Java diffèrent. Une application Java est une entité indépendante qui va s'exécuter dans l'environnement de l'interpréteur. L'application n'a aucun contexte graphique (Frame). Il faut en créer un. Une applet est un programme Java compilé qui est exécuté par le logiciel AppletViewer* ou un navigateur WWW qui supporte Java.

Les caractéristiques de base des applets en font un outil idéal pour l'Internet. En effet, celles-ci sont petites, peuvent être téléchargées et exécutées très rapidement.

Elles ajoutent un composant interactif et fonctionnel aux pages WWW. Par contre elles ont un accès limité aux ressources réseaux et aux systèmes de fichiers locaux pour des raisons de sécurité. Le bytecode compilé est le composant exécutable d'une applet qui est intégrée dans une page au format HTML et est automatiquement téléchargée sur le poste client. Elle est ensuite exécutée par celui-ci.

2.2. Filiation de la classe Applet

```
java.lang.Object
  +—java.awt.Component
    +—java.awt.Panel
      +—java.applet.Applet
```

3. Intégration de l'applet dans une page Web

Il existe des balises (tags) HTML spécifiques pour indiquer au browser qu'une applet est présente dans une page. Il s'agit de <Applet>. Voici la structure du fichier HTML :

```
<html>
opplet codebase=EndroitOuTrouverLeCode code=NomDuFichier.class
Width=largeur height=hauteur>
  <param name="Paramètre1" value="ValeurDuParamètre"
  <param name="Paramètre2" value="ValeurDuParamètre"

  <param name="ParamètreN" value="ValeurDuParamètre"
</applet>

</html>
```

L'applet sera donc exécutée par le navigateur Web à l'endroit où se trouve le tag (applet) avec les paramètres physiques (width, height...) spécifiés et logiques (paramètre 1, ..., paramètre).

4. Mise en oeuvre d'une applet

Tout programme Java est une classe. Une applet n'échappe pas à cette règle. Si on veut créer une applet, on doit étendre la classe `java.applet.Applet`. Cette classe contient les méthodes nécessaires à la gestion de l'applet, et à l'interaction de celle-ci avec son environnement (browser).

L'applet simple, comme toutes les autres applet, est une extension de la classe `Applet`. Le code peut se résumer à la surcharge de quatre (04) méthodes, pour traiter les événements! 1]:

```
public void init ()
```

Le browser fait appel à cette méthode lorsque l'applet est chargée (ou rechargée).

Cette méthode devra charger les informations telles que des images ou sons, et récupérer les paramètres présents dans la page HTML.

```
public void start ()
```

Après avoir été initialisée, l'applet est démarrée, grâce à la méthode `start()`. L'applet est également redémarrée après avoir été stoppée, lorsqu'elle est à nouveau visible.

```
public void stop ()
```

Cette méthode permet à l'applet de s'arrêter lorsqu'elle n'est plus visible, parce que l'utilisateur a changé de page, par exemple.

```
public void destroy ()
```

L'applet est détruite lorsque le browser s'arrête, ou avant que l'applet soit rechargée. Cette méthode doit être remplacée si on veut stopper des threads créés par `start()` de l'applet.

```
public void paint(Graphics g)
```

Cette méthode est appelée chaque fois que l'on doit redessiner l'applet. Le paramètre est de type `Graphics` et représente la surface de dessin sur laquelle on doit travailler.

5. La chaîne d'exécution - de la conception à l'exécution

La première étape consiste à écrire le programme en JAVA (fichier.java).

Ensuite il faut le pré-compiler avec le compilateur `javac` (lui-même écrit en java et donc portable sur toutes machines). Celui-ci génère un fichier contenant le P-code (fichier.class). C'est ce P-code qui circulera sur le réseau.

Arrivé sur la machine hôte, le P-Code est pris en charge par le Run-Time JAVA. Dans un premier temps, le byte code vérifié se charge d'assurer l'intégrité de la portion de code reçue (contrôle des E/S, manipulation de la mémoire, etc.). Dans un deuxième temps, Class Loader charge chaque classe importée dans un espace d'adressage distinct et les classes locales dans un seul espace d'adressage. Puis enfin l'interpréteur permet d'exécuter l'application.

6. La Sécurité dans Java[5].

Conçu pour exécuter des programmes transmis par le réseau Internet, Java nécessite des mécanismes de sécurité renforcés. L'interpréteur Java vérifie chaque programme avant de le lancer, si nécessaire avec un algorithme d'authentification à clé publique, et l'empêche d'accéder au disque local ou au réseau sans autorisation expresse.

6.1. Allocation et représentation de la mémoire

Une des premières lignes de défense du compilateur Java réside dans son allocation mémoire et dans son modèle de référence. Avant tout, les décisions de représentation de la mémoire ne sont pas prises par le compilateur Java comme en C ou en C++. En fait, la représentation de la mémoire est différée à l'exécution et sera différente selon les caractéristiques des plates-formes sur lesquelles le système Java s'exécute.

De plus Java, n'a pas de pointeurs vers des cellules de mémoire contenant les adresses d'autres cellules mémoire. Le code Java compilé référence la mémoire via des références symboliques qui sont converties lors de l'exécution par l'interpréteur JAVA en adresses mémoires réelles. Les programmeurs en Java ne peuvent pas créer de pointeurs vers la mémoire car l'allocation et le modèle de référence sont entièrement gérés par le système runtime.

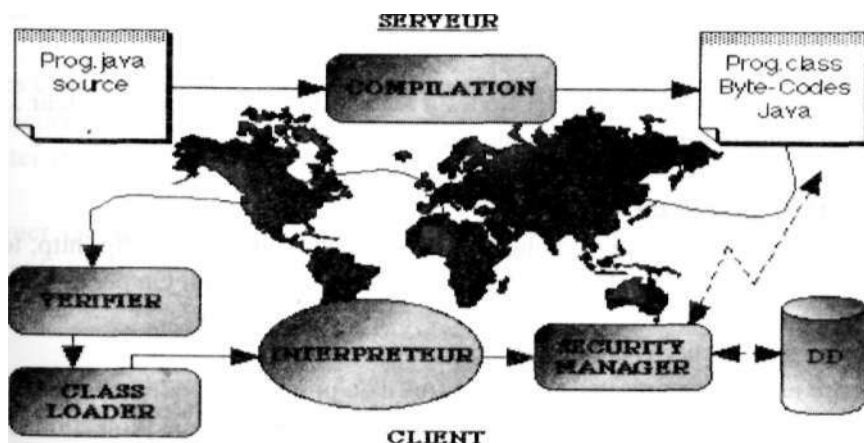


Figure 1 - La sécurité avec Java

6.2. Processus de Vérification des Bytes codes

Le compilateur Java effectue un contrôle étendu et rigoureux lors de la compilation, de manière à détecter très tôt les erreurs de syntaxe. De plus les déclarations de classes et des méthodes doivent être explicites, ce qui accroît le degré de sécurité.

Le système runtime Java vérifie tout le code importé en le soumettant à une vérification: on parle de byte code vérification.

Des tests permettent de vérifier que le format d'un fragment de code est correct, c'est-à-dire qu'il respecte les règles suivantes :

- Il ne crée pas de pointeurs.
- Il ne viole pas les restrictions d'accès.
- Il accède aux objets tels qu'ils sont.

6.3 Vérificateur de byte code

Le vérificateur de byte code parcourt les bytes codes, construit l'information sur l'état du type et vérifie le type des paramètres de toutes les instructions byte code.

Quelle que soit l'origine du source, le chargeur de byte codes et le vérificateur de bytes codes effectuent systématiquement un contrôle du code avant l'exécution.

Le vérificateur de byte code agit comme un gardien : il assure que le code passé à l'interpréteur est dans un état correct pour être exécuté. Une fois que la vérification est effectuée, le code satisfait les propriétés suivantes :

- Pas de débordement ou de sous-débordement de piles
- Le type des paramètres de toutes les instructions en byte codes est connu pour être toujours correct.
- Les champs d'accès des objets sont connus pour être légaux (private, public ou protected).

Une fois que le vérificateur de bytes codes a achevé son travail, l'interpréteur Java peut procéder de manière rapide car il n'a plus aucun contrôle à effectuer.

6.4. Sécurité au niveau du réseau

Java fournit des interfaces pour manipuler les différents protocoles (ftp, http, telnet,...). Cela constitue la ligne de défense au niveau de l'interface réseau. On peut ainsi :

- Empêcher tous les accès au réseau
- Permettre les accès au réseau aux seuls sites d'où provient le code importé
- Permettre les accès au réseau seulement en dehors d'une zone de protection si le code provient de l'extérieur
- Permettre tous les accès au réseau.

7. Ce que l'applet peut faire ?

- Une applet peut communiquer avec d'autres applets ou avec le browser.
- Applets : en invoquant les méthodes publiques d'autres applets de la même page.
- Browser : En communiquant avec le browser : Java offre un ensemble de primitivcq limité pour assurer une communication entre l'applet et le browser qui l'exécute. La plus importante est celle qui permet de communiquer au browser quelle URL afficher (ShowDocument de AppletContext).

8. Ce que l'applet ne peut pas faire ?

Chaque browser implémcnte sa propre politique de sécurité pour éviter que les applets ne provoquent de dégâts. Actuellement, les browsers imposent les' restrictions suivantes pour toute applet chargée dans le réseau :

- Une applet ne peut ni lire ni écrire sur le site hôte (elle deviendrait dans le cas contraire un vecteur idéal pour les virus).
- Une applet ne peut pas lire certaines propriétés du système
- Une applet ne peut pas charger des bibliothèques ou définir des méthodes natives
- Une applet ne peut pas établir des connexions réseau sauf pour le site dans lequel elle se trouve.
- L'applet ne peut exécuter aucun programme en dehors de son site d'origine.

Lorsqu'un browser détecte une violation de ces restrictions, une exception est générée. L'applet capte cette exception et réagit en conséquence.

Malgré ces restrictions, de nombreuses utilisations pour les applets Java sont réalisées :

- Visualiser des molécules en trois dimensions en les faisant pivoter à la souris
- Naviguer dans l'énorme quantité d'images d'un corps humain découpé tranche par tranche.

Suivre les cours de la bourse en direct, et voir se recalculer le montant de vos actions au fil des fluctuations.

- Présenter de la publicité interactive, permettant de gérer des profils d'usagers très précis.
- Dialoguer en direct sur une page Web, par écrit, avec toutes les personnes qui consultent la même page Web au même moment que vous.

Références Bibliographiques ¶¶¶

[1]. "The Java Tutorial :Object Oriented Programming for the Internet"

M. Campionc, K. Walrath

1997

<http://java.sun.com/nav/read/Tutorial>

[2]. "Teach yourself Java in 21 days"

L. Lemay, CL. Perkins.1996

<http://www.lne.com/Web/Java>

[3]"The Java Saga"

D. Bank

1996

<http://www.hfd.hr/Student/clanci/java.html>

[4]"Java: la révolution logicielle"

Institut Prfométhcus, 1996

<http://www.prometheus.eds.fr/actu/article>

[5]"Security in Java" White Paper

<http://arrakis.ucd.ie/~paulmc/java>

[6]"Introduction to Java "White Paper

[http:// arrakis.ucd.ie/~paulmc/java](http://arrakis.ucd.ie/~paulmc/java)