

## SERVICES WEB & Web des SERVICES

Francis.Lapique@epfl.ch, e-pfl

### INTRODUCTION

Si la mutualisation des *services Web* au niveau inter-entreprises ou plus généralement dans le Web des services est encore musique d'avenir, au niveau de l'entreprise elle est d'ores et déjà en marche. Pour quelle raison? Parce que la maîtrise des coûts liés aux problèmes de l'interopérabilité des services métiers passe par l'affranchissement de l'intégration d'environnements *middleware* propriétaires. C'est un tournant important que l'EPFL comme toute entreprise devra négocier au mieux dans le cadre de sa politique de systèmes d'information.

Les approches techniques pour aborder l'intégration des services sont pléthoriques. Sur le marché de l'*Open Source*, j'ai décidé d'explorer trois démarches:

- SOAP (*Simple Object Access Protocol*), une spécification de communication entre services Web sur une base d'échange de messages XML
- JXTA, un réseau pair-à-pair (*Peer-to-Peer*)
- Jahia, un serveur d'applications sur la plate-forme J2EE.

Qui dit services dit clients de ces services. C'est le point de vue de cet article. Le côté serveur, par ses aspects

trop spécifiques, n'est pas abordé ou très peu. Pour rendre l'exercice le plus didactique possible, tous les codes sources sont en Java pour les programmeurs qui voudraient expérimenter ce que je décris. Les trois chapitres SOAP, JXTA et Jahia sont indépendants.

Avant de rentrer dans le vif du sujet, quelques mots tout de même du service.

### LE SERVICE

Le service **exemple** qui va nous accompagner tout au long de cet article n'est pas un service d'achats, de réservation ni de recherche de vie extra-terrestre mais celui d'un automate à l'affût des dernières nouvelles d'un certain nombre de sujets à caractères scientifiques. Sans rentrer dans trop de détails techniques, qui dépasseraient le cadre de cet article, on dirige l'automate sur un ensemble de sites, qui va catégoriser des documents sur la base d'un modèle d'apprentissage. Les méthodes d'apprentissage sur des données textuelles sont nombreuses et les limites de chacune bien connues. Pour ceux que cela intéresse voici quelques mots-clés pour découvrir une littérature,

**SUITE EN PAGE 8**

## SOMMAIRE FI 8/2002

Services Web & Web des services <i>Francis Lapique</i>	1
Libre-service mais sans abus <i>Jacques Virchaux</i>	2
XSLT et XPath au service de vos données XML <i>Guy de Pourtalès</i>	3
Les enjeux du référencement des locaux <i>Philippe Pichon</i>	7
Programme des cours	15
Flash sur Flash ... dans le Flash ! <i>Hicham Dennaoui</i>	19
Calendrier	20

### PROCHAINES PARUTIONS

	délaI	rédaCtion	paraution
9	31.10.02		19.11.02
10	28.11.02		17.12.02

# Libre-service MAIS SANS ABUS

Jacques.Virchaux@epfl.ch, SIC



La mise en service de points d'accès sans fil, conjointement aux prises (ou câbles) jaunes, offre un confort sans cesse croissant, principalement aux étudiants qui peuvent utiliser leur PC portable en de nombreux endroits (<http://network.epfl.ch/alise>) en libre service. L'accès au réseau de l'EPFL étant réalisé par un tunnel VPN assurant la confidentialité des données, l'utilisateur est alors authentifié par son numéro SCIPER qui figure sur la carte CAMIPRO et son mot de passe GASPARE qui aura été demandé sur une borne OSCAR.

Après avoir constaté quelques rares abus, notamment avec le protocole P2P (téléchargement de fichiers très volumineux de type MP3 ou DivX), il est bon de rappeler les directives existantes auxquelles le personnel et les étudiants sont soumis à l'EPFL: <http://sicwww.epfl.ch/informatique/directives.html>. Actuellement, l'EPFL paie au volume le trafic entrant qui a transité par la ligne transatlantique. Par exemple, un transfert d'un Gigabyte représente un montant de Fr. 20.- qui est payé à SWITCH. Si cela peut paraître cher par rapport aux abonnements individuels proposés par les fournisseurs, il faut aussi savoir que la bande passante n'est pas la même.

Pour rappel, le protocole P2P est principalement utilisé par des programmes de type KaZaA, eDonkey ou iMesh. Ces logiciels comportent une partie serveur permettant d'offrir à l'ensemble de la communauté Internet des fichiers pour le téléchargement. Très souvent, ils ont aussi un petit espion qui permet de tracer les recherches et les fichiers téléchargés pour obtenir un profil d'utilisateur qui peut ensuite être revendu aux spammeurs pour une publicité bien ciblée ! Pire encore, ils peuvent dévoiler à votre insu beaucoup plus d'informations que ce qu'on désire partager (c.f. article récent en anglais: <http://zdnet.com/2100-1107-960638.html>).

En outre, la Loi sur les droits d'auteurs (LDA) est précise concernant les copies ou mises à disposition d'œuvres sans le consentement du créateur: [http://www.admin.ch/ch/fr/rs/c231\\_1.html](http://www.admin.ch/ch/fr/rs/c231_1.html).

Merci d'observer un comportement responsable d'adulte averti et d'éviter ainsi toute mesure désagréable que des abus peuvent déclencher, ceci sans compter la possibilité de voir ternie l'image de marque de l'EPFL. ■



## LE SIC EN BREF



### Service informatique central

<http://sic.epfl.ch/>

e-mail: [secretariat.sic@epfl.ch](mailto:secretariat.sic@epfl.ch)

Bâtiment des Mathématiques et Pavillon C  
EPFL-Ecublens  
CP 121 - 1015 Lausanne

+41 21 69 32211 - fax +41 21 69 32220

*les photos manquantes le sont à la demande des intéressés*

mise en page: Appoline - 01.10.02

Vient de paraître, demandez-le à la réception du SIC

## Flash informatique

Les articles ne reflètent que l'opinion de leurs auteurs. Toute reproduction, même partielle, n'est autorisée qu'avec l'accord de la rédaction et des auteurs.

Rédacteur en chef: Jacqueline Dousson, [fi@epfl.ch](mailto:fi@epfl.ch)  
Mise en page & graphisme:  
Appoline Raposo de Barbosa

Comité de rédaction: Omar Abou Khaled, Jean-Daniel Bonjour, Nicolas Bouche, Milan Crvcenin, Pierre Kuonen, Jacques Menu, Elaine Mc Murray, Philippe Pichon, François Roulet, Christophe Salzmänn & Jacques Virchaux  
Atelier de Reprographie EPFL

Impression:

Tirage: 4000 exemplaires  
Adresse Web: <http://sic.epfl.ch/publications/>

Adresse: SIC-SA EPFL, CP 121,  
CH-1015 Lausanne  
Téléphone: +41 21 69 32246 & 32247

# XSLT ET XPATH

## AU SERVICE DE VOS DONNÉES XML

Guy de Pourtalès, Architecte EPFL, Chef de projet Serpentine, depourtales@vtxnet.ch

### INTRODUCTION

XML, XSLT, XPath, XQuery, SVG... bien des noms qui monopolisent les conversations de tout informaticien un peu branché. Mais, au fond, à quoi tout cela peut bien servir ?

Cet article n'a aucunement l'ambition de faire de vous des maîtres du «X+quelque-chose». La seule prétention qu'il a, peut-être, est de montrer comment l'intégration de plusieurs de ces techniques peut vous aider à manipuler des données et à les représenter selon les besoins, ceci de manière simple et avec des outils aussi standards que Internet Explorer.

Malgré cela, vous aurez tout de même droit à un bref historique du XML, ainsi que les bases de sa définition. Après cette introduction et la présentation des outils utilisés pour les explications, un exemple pratique (et très basique) de transformation d'un XML (*eXtensible Markup Language*) en un SVG (*Scalable Vector Graphics*) servira de guide pour la présentation de fonctions XSL (*eXtensible Stylesheet Language*).

### BREF HISTORIQUE

En 1986 le premier pas vers une description généralisée de la structure des données apparaît sous la forme du SGML (*Standard Generalized Markup Language*), norme ISO 8879:1986. Cette norme définit en effet un langage qui permet de définir de manière flexible et cohérente l'organisation et la structuration de données textuelles et binaires: la DTD (*Document Type Definition*). Celle-ci apporte un concept essentiel au sujet qui nous préoccupe: la structuration des données par balises. Les premiers utilisateurs ont été en particulier les bibliothèques qui ont pu constituer et échanger ainsi des catalogues d'ouvrages importants. Mais l'apparente complexité de son utilisation et de sa mise en place n'a pas trouvé un large écho.

Neuf ans après, soit en 1995, le mythique HTML (*Hypertext Markup Language*) voit le jour. A l'aide d'une DTD SGML, le HTML définit un langage accessible permettant la représentation ordonnée de données. Le HTML ajoute également un concept clé: les liens hypertextes, qui font le bonheur de tous les internautes frénétiques. Dans tout traitement de texte, le concept de modèle apporte un gain de temps pour la mise en page, mais surtout assure une cohérence graphique entre les documents produits. Les sites créés en HTML devenant de plus en plus importants et monopolisant de plus en plus de personnes pour leur gestion, il fallait un équivalent au HTML. C'est pourquoi, en 1996, les CSS (*Cascading Stylesheets*) vont venir compléter les outils à disposition.

Le HTML est un langage qui permet de structurer les données, mais de manière essentiellement graphique. La structure des données devient en effet évidente une fois affichée correctement dans le visualiseur adéquat. Pour répondre à ce manque (qui n'en était pas réellement un, au vu de l'existence du SGML!), le W3C (*World Wide Web Consortium*) a édité la recommandation XML en 1998. Dans sa première version, le XML utilise en effet la DTD pour définir la structure obligatoire d'un document. Toutefois, afin que cette structure puisse être définie en XML, le W3C a développé le XMLSchema appelé à remplacer la DTD.

Comme le HTML bénéficie des CSS pour la création de modèles cohérents et réutilisables, le W3C a introduit le XSLT (*eXtensible Stylesheet Language for Transformations*) qui permet de transformer les données d'une structure XML à une autre. XSLT est indissociable de XPath qui définit les règles d'accès et de navigation dans une structure XML. Comme nous le verrons plus loin, XPath s'intègre directement dans XSLT.

Finalement, en 2001 apparaît la spécification XSL (*eXtensible Stylesheet Language*) qui réunit en son sein XSLT, XPath et XSL:FO (*eXtensible Stylesheet Language: Formatting Objects*). Ce dernier langage répond à la nécessité de compléter la chaîne de processus de sélection, transformation et présentation de données structurées. XSLT et XPath, conjointement, permettent de sélectionner et modéliser des données alors que XSL:FO permet de représenter les résultats dans une mise en page prédéfinie. Actuellement, le format de sortie privilégié de XSL:FO est le PDF (*Portable Document Format*).

Il faut noter qu'à l'heure actuelle aucun moteur de traitement ne supporte entièrement et strictement la recommandation XSL. Toutefois, certains moteurs ajoutent des fonctionnalités supplémentaires de leur cru.

En fin d'article, vous trouverez les liens vers les spécifications valables à l'heure de la rédaction de cet article.

### COMMENT FONCTIONNE XSL

A partir de maintenant, nous allons nous concentrer sur XSLT et XPath et faire l'impasse, à dessein, sur XSL:FO, qui mérite un article complet à lui seul. Ceci étant dit, le fonctionnement d'une transformation ressemble à n'importe quel processus de traitement de données. Un moteur XSL reçoit un fichier XML en entrée, un fichier XSLT en paramètre et produit un ou plusieurs fichiers XML, chaque fichier XSLT peut faire appel à d'autres fichiers XSLT ou inclure des fichiers XML supplémentaires, comme le montre le schéma 1.

Les fonctions XSLT vous seront présentées au fur et à mesure de notre démonstration. Mais avant de se lancer dans

notre première transformation, il faut passer par une brève présentation de ce qu'est XPath (et accessoirement, à quoi il sert!).

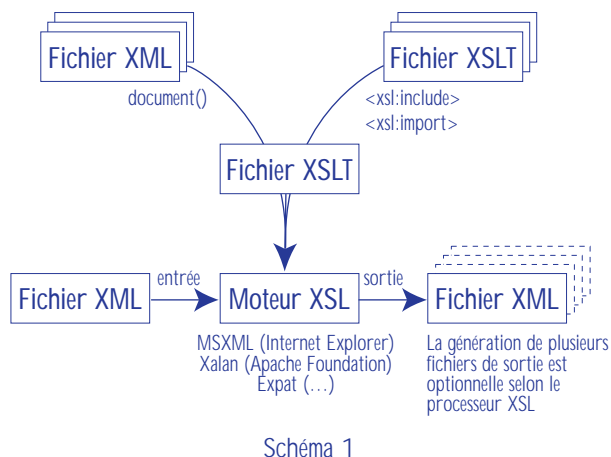


Schéma 1

Pour se repérer dans une structure XML depuis n'importe quel emplacement dans l'arbre (contexte), XPath définit six axes d'accès comme le montre le schéma ci-après. A l'aide des références à ces axes, vous pouvez accéder à n'importe quelle partie de votre structure XML.

namespace  
(espace de nommage)

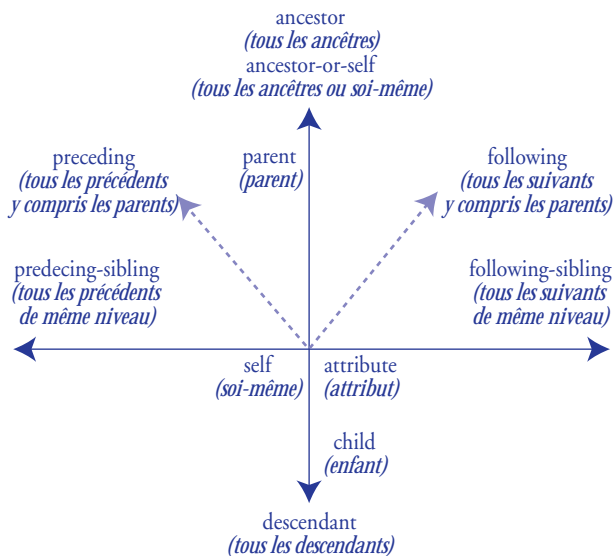


Schéma 2

En outre, XPath offre une série de fonctions de calcul, de traitement et de sélection des nœuds fort utiles dont nous utiliserons une partie par la suite.

Avant de procéder à une première réelle transformation, nous allons étudier une première transformation XSLT qui produit invariablement le même résultat, quel que soit le fichier XML traité:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl=
  "http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html"/>

  <xsl:template match="/">
    <html><body>
      Une transformation inutile
    </body></html>
  </xsl:template>
</xsl:stylesheet>
```

Le résultat sera toujours le document XHTML (pas très passionnant) suivant:

```
<html><body>
Une transformation inutile
</body></html>
```

Malgré son apparente inutilité, la définition de la XSLT permet de poser quelques règles de base:

**ligne 1:** un document XSLT est un document XML et doit comporter cet en-tête

**ligne 2:** un document XSLT définit toujours une balise racine `<xsl:stylesheet>` dont la référence de l'espace de nommage (`xmlns:xsl`) est située sur le site du Consortium W3.

**ligne 3:** un document XSLT peut définir le type de document qu'il produit. Ceci est important dans le cas de traitements nécessitant la définition du type MIME du document résultant.

**ligne 5 à 9:** un document XSLT définit des méthodes (`<xsl:template>`) appliquées à des nœuds. Dans ce cas, la méthode est appliquée au nœud racine («/»).

**ligne 6 à 8:** toutes les balises et les textes situés hors des balises de l'espace de nommage «xsl:» sont dirigés vers le document de sortie.

En résumé, notre XSLT a le comportement suivant: pour le nœud racine rencontré, écrire les balises HTML et le texte.

## DANS LE VIF DU SUJET

Le système Serpentine est un transport public entièrement automatisé constitué de trois composants principaux: les capsules (véhicules), la piste en réseau (transmission d'énergie, guidage) et le système de gestion (*Traffic Manager HB®*). Ce dernier assure l'automatisation des déplacements individuels d'un ensemble de véhicules selon la demande des clients et la répartition de la charge sur le réseau. Le système de gestion s'appuie sur un grand nombre de données au format XML qui nourrissent les algorithmes sous-jacents. Dans le cas de simulations pour le dimensionnement des réseaux ou dans celui de la surveillance des mouvements des capsules, il est indispensable de représenter ces données et les résultats des calculs sous forme graphique. Pour ce faire, nous utilisons une XSLT qui produit un fichier SVG. Nous utilisons Xalan de la Fondation Apache comme moteur XSL, en particulier dans le cadre d'un serveur Jakarta Tomcat avec Cocoon comme gestionnaire de transformation XSL. Pour visualiser les SVG, nous utilisons soit le plug-in SVGViewer d'Adobe, soit le paquet Batik de la Fondation Apache.

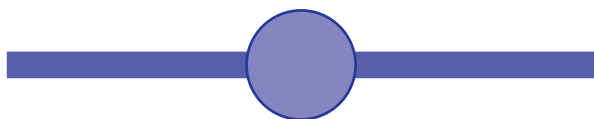
Jetons tout d'abord un coup d'œil sur un extrait des données XML que nous aurons à traiter:

```
<?xml version="1.0"?>
<MELFIN>
  <ELFIN ID_G="200209292309436"
  ID="200209301053265">
    <FORME>
      <GEOGRAPHIE>
        <POINT>0.0,0.0</POINT>
        <POLYLIGNE>
          1,-10.0,0.0
          2,10.0,0.0
        </POLYLIGNE>
      </GEOGRAPHIE>
    </FORME>
```

```
</ELFIN>
</MELFIN>
```

L'objectif de cet article est de montrer comment récupérer la coordonnée X du point pour la placer comme centre du cercle représentant le point.

Venons en au du SVG modèle que nous devons obtenir:



```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20001102/
/EN" "http://www.w3.org/TR/CR-SVG-20001102/DTD/
svg-20001102.dtd">
<svg>
  <g transform="translate(0,0)" style="fill:red;
stroke:blue; stroke-width:01;">
    <circle cx="0.0" cy="0.0"
            r="2" id="200209301053265"/>
    <polyline style="stroke-width:1;"
            points="-10.0,0.0 10.0,0.0"/>
  </g>
</svg>
```

La coordonnée X récupérée doit se placer comme valeur de l'attribut cx de la balise <circle>.

Maintenant que la source et la cible sont identifiées, nous pouvons commencer à construire notre XSLT:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl=
" http://www.w3.org/1999/XSL/Transform">

  <xsl:output="xml"
doctype-public=" "-//W3C//DTD SVG 20001102//EN"
doctype-system=" http://www.w3.org/TR/CR-SVG-
20001102/DTD/svg-20001102.dtd"/>

  <xsl:template match="/"/>
</xsl:stylesheet>
```

Dans cette première étape, nous définissons le type de document généré, soit un document XML répondant à la DTD SVG référencée sur le site du W3C. Toutefois, notre XSLT ne traite pour l'instant aucune donnée.

Afin de définir les balises principales du SVG, nous faisons simplement correspondre les balises principales du SVG (soit <svg> et <g>) à la balise racine de notre XML:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl=
" http://www.w3.org/1999/XSL/Transform">
  <xsl:output="xml"
doctype-public=" "-//W3C//DTD SVG 20001102//EN"
doctype-system=" http://www.w3.org/TR/CR-SVG-
20001102/DTD/svg-20001102.dtd"/>
  <xsl:template match="/">
    <svg>
      <g transform="translate(0,0)" style=
" fill:red;stroke:blue;stroke-width:01;">
        </g>
      </svg>
    </xsl:template>
  </xsl:stylesheet>
```

Le résultat comme attendu définira les balises SVG telles quelles dans le document de sortie:

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20001102/
/EN" "http://www.w3.org/TR/CR-SVG-20001102/DTD/
svg-20001102.dtd">
```

```
<svg>
  <g transform="translate(0,0)" style=
" fill:red; stroke:blue; stroke-width:01;">
    </g>
</svg>
```

Comme l'élément (la coordonnée X) qui nous intéresse se trouve dans une balise <POINT>, nous allons définir une méthode s'appliquant spécifiquement à cet élément et y faire appel depuis le traitement du noeud racine:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0" xmlns:xsl=
" http://www.w3.org/1999/XSL/Transform">
  <xsl:output="xml" doctype-public=" "-//W3C//DTD
SVG 20001102//EN"
doctype-system=" http://www.w3.org/TR/
CR-SVG-20001102/DTD/svg-20001102.dtd"/>
  <xsl:template match="/">
    <svg>
      <g transform="translate(0,0)"
style="fill:red; stroke:blue;
stroke-width:01;">
        <xsl:apply-templates select="MELFIN/
ELFIN/FORME/GEOGRAPHIE/POINT">
          </g>
        </svg>
      </xsl:template>

      <xsl:template match="MELFIN/ELFIN/FORME/
GEOGRAPHIE/POINT">
        <circle cx="2" cy="2" r="2"/>
      </xsl:template>
    </xsl:stylesheet>
```

Une méthode peut s'appliquer spécifiquement à un noeud de l'arbre XML. Son accès s'effectue toujours relativement au contexte dans lequel se trouve le processeur. Dans notre cas, comme nous voulons appeler la méthode depuis le contexte racine du traitement, nous devons spécifier le chemin complet jusqu'à la balise qui nous concerne. Ceci est une première application de XPath. Notez que, volontairement, les valeurs cx et cy sont fausses.

L'appel à une méthode s'effectue à l'aide de l'ordre <xsl:apply-templates select="»chemin/du/noeud/relatif/au/contexte»>. Si le noeud <POINT> n'existait pas, l'appel ne s'effectuerait simplement pas.

Voici donc le résultat:

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20001102/
/EN" "http://www.w3.org/TR/CR-SVG-20001102/DTD/
svg-20001102.dtd">
<svg>
  <g transform="translate(0,0)" style="fill:red;
stroke:blue; stroke-width:01;">
    <circle cx="2" cy="2" r="2"/>
  </g>
</svg>
```

Dans le cas qui nous préoccupe, nous voulons modifier la valeur de l'attribut cx de la balise <circle>, c'est pourquoi nous devons sortir l'attribut de la balise afin de pouvoir lui assigner une valeur:

```
<xsl:template match="MELFIN/ELFIN/FORME/
GEOGRAPHIE/POINT">
  <circle cy="2" r="2">
    <xsl:attribute name="cx">3</xsl:attribute>
  </circle>
</xsl:template>
```

La modification dans le document est invisible, à part le changement de la valeur de `cx` de 2 à 3, preuve que l'appel à `<xsl:attribute>` a fonctionné.

Vous vous souvenez que les coordonnées `x` et `y` sont séparées par une virgule dans la définition du point de notre document source. Afin d'extraire la coordonnées `x`, nous allons devoir créer une méthode **nommée** dont la fonction sera d'extraire tous les caractères jusqu'au premier séparateur passé en paramètre. Elle recevra également en paramètre la valeur de la balise `<POINT>` soit le vecteur «`x,y`».

Ajoutons à notre XSLT la méthode suivante:

```
<xsl:template name="getPointX">
  <xsl:param name="coordinates"/>
  <xsl:param name="separator"/>
  <xsl:value-of select="substring-
    before($coordinates,$separator)"/>
</xsl:template>
```

Les paramètres des méthodes sont définis au début de la méthode par `<xsl:param>`. Pour faire référence à un paramètre ou une variable dans une fonction XSLT ou XPath, nous utilisons la convention `$nom_de_la_variable`.

L'ordre `<xsl:value-of>` a pour fonction d'écrire en sortie la valeur choisie par l'attribut `select`. Si, par exemple, nous avons écrit `<xsl:value-of select="MELFIN/ELFIN/FORME/GEOGRAPHIE/POINT">`, nous aurions obtenu à la sortie «0.0,0.0». Afin d'extraire le texte avant un séparateur, nous faisons appel à une fonction XPath `substring-before(texte,texte)`.

Afin d'être une dernière fois didactique, nous allons définir une variable dans notre méthode de traitement de la balise `<POINT>` qui prendra la valeur résultante de l'appel à la méthode `getPointX`:

```
<xsl:template match="MELFIN/ELFIN/FORME/
  GEOGRAPHIE/POINT">
  <xsl:variable name="x">
    <xsl:call-template name="getPointX">
      <xsl:with-param name="coordinates">
        <xsl:value-of select="."/>
      </xsl:with-param>
      <xsl:with-param name="separator">,
      </xsl:with-param>
    </xsl:call-template>
  </xsl:variable>

  <circle cy="2" r="2">
    <xsl:attribute name="cx">
      <xsl:value-of select="$x"/></xsl:attribute>
    </circle>
  </xsl:template>
```

La définition d'une variable s'effectue à l'aide de l'ordre `<xsl:variable>`, sa valeur étant définie entre la balise d'entrée et de fin. Afin de récupérer la valeur `x` que nous cherchons, nous appelons la méthode `getPointX` à l'aide `<xsl:call-template>` à laquelle nous passons la valeur du contexte actuel (rappelez-vous: `<POINT>`). XPath définit le contexte actuel par «.» tout comme sur un système de fichier UNIX. En plus, nous passons une virgule comme paramètre de séparation. Remarquez que les paramètres sont nommés lors de l'appel à la méthode, l'ordre de passage n'important pas.

Pour finir, nous passons à la valeur de l'attribut `cx` la valeur de la variable `x`, pour obtenir le document suivant:

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 20001102/
/EN" "http://www.w3.org/TR/CR-SVG-20001102/DTD/
svg-20001102.dtd">

<svg>
  <g transform="translate(0,0)" style="fill:red;
    stroke:blue; stroke-width:01;">
    <circle cx="0.0" cy="2" r="2"/>
  </g>
</svg>
```

## Conclusion

Le petit exemple qui a été développé ci-dessus est très loin d'être optimal. Pour être franc, il pourrait être beaucoup plus compact, mais il a été écrit avec en arrière pensée la volonté de montrer quelques possibilités de XSLT et de XPath.

XSLT et XPath sont des langages très riches et les possibilités de traitement de structures XML étendues. Les notions de tri, de copie, de prédication, de recherche, de boucle, de traitement conditionnel, etc. n'ont pas été évoquées car beaucoup trop nombreuses pour entrer dans le cadre de cet article. Le meilleur conseil est d'entrer dans une librairie ou une bibliothèque et de prendre les quelques livres consacrés au sujet. En particulier, les éditions O'Reilly proposent des ouvrages très clairs et particulièrement bien structurés.

Voici encore quelques sites qui pourront vous être utiles lors de votre appréhension de ce nouveau type de traitement de données.

## LES RECOMMANDATIONS DU W3C

SGML: <http://www.iso.ch/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=16387>  
 HTML: <http://www.w3.org/TR/html4/>  
 XHTML: <http://www.w3.org/TR/xhtml11/>  
 CSS: <http://www.w3.org/TR/REC-CSS2/>  
 XML: <http://www.w3.org/TR/REC-xml>  
 XSLT: <http://www.w3.org/TR/xslt20/>  
 XPath: <http://www.w3.org/TR/xpath20/>  
 XSL (avec XSL:FO): <http://www.w3.org/TR/xsl/>

## LES SITES

- Le World Wide Web Consortium: <http://www.w3.org/>
- La Fondation Apache et leurs outils Open Source dédiés à XML: <http://xml.apache.org>
- Des didacticiels en nombre: <http://www.w3schools.com>
- Un portail d'information riche: <http://www.xml.com>
- Une collection de lien vers des outils XML gratuits: <http://www.garshol.priv.no/download/xmltools>
- Le système Serpentine <http://www.serpentine.ch> ■

# LES ENJEUX du RÉFÉRENCIEMENT DES LOCAUX

Philippe.Pichon@epfl.ch, SIC



Vous avez pu lire dans le Flash Informatique numéro 7/2002 l'article de Claude Philipona au sujet du système d'aide à l'orientation sur le site de l'EPFL (<http://plan.epfl.ch>). A ce propos, avez-vous essayé d'intégrer dans votre signature de messagerie électronique ou votre site Internet un lien vers le plan de votre bureau, tel que celui-ci: Vous pouvez me trouver ici: <http://plan.epfl.ch?room=PC026?> Voilà de quoi aider vos visiteurs à vous trouver plus rapidement. Pour plus de détails, je vous renvoie à l'article précité.

Mais mon propos aujourd'hui est de vous présenter une partie importante des pièces qui constituent le puzzle d'informations fourni à cet outil: la base de données des locaux, avec son applicatif de gestion développé à la section Informatique de gestion du SIC pour le compte du Service des bâtiments, j'ai nommé Locaux2. Car il n'y a pas de navigation virtuelle sans données bien réelles.

Quels sont les volumes? Evidemment, gérer quelques bureaux et salles de cours pourrait se faire à la main. Mais pour l'EPFL, il ne s'agit pas moins de 10'000 espaces, répartis en 5 principaux sites pour une petite centaine de bâtiments. Cela va du bureau présidentiel à l'observatoire de la Jungfrauoch en passant par le Château de Bassenges et les salles d'enseignement. Il faut, en effet, savoir que sont répertoriés des espaces aussi divers que les ascenseurs, auditorios, gaines techniques, escaliers, laboratoires, restaurants, abris, secrétariats, etc. En fait, il y a actuellement plus de 1'000 types d'utilisation différents !

Gérer les données des locaux n'est donc pas une mince affaire. Sans encore parler de facturation, il faut déjà les référencer: ils ont donc tous un numéro d'identification unique, attribué automatiquement par le SGBD. C'est ce fameux numéro d'identification (que l'on pourrait appeler SCIOLOC, par analogie au SCIPER), qui peut alors être repris dans d'autres bases de données, par exemple celle des 30'000 prises du réseau informatique ou le bottin téléphonique, et qui sert à consolider facilement et sûrement les données relatives à un local, bien qu'elles proviennent de diverses sources.

Si chaque local dispose d'un SCIOLOC unique, toujours fixe et destiné aux ordinateurs (le SCIOLOC du local PC 026 est le numéro 6325), il possède également un code unique pour l'œil humain (ce code peut changer en de rares occasions, par exemple si l'on renomme un bâtiment). Le code du local est constitué automatiquement en fonction de l'une des 5 règles de nommage spécifique décidées par bâtiment, règle qui peut inclure ou non l'abréviation des bâtiments, zone, étage et un numéro de local, ce qui permet de tenir compte de la plaquette d'identification déjà apposée sur la porte.

Bien entendu, l'EPFL n'a pas attendu 2002 pour gérer ses locaux: la toute première version date de 1987 et tournait sur une machine Data General. Les données de l'épo-

que étaient déjà utilisées pour des réquisitions de travaux, des locations de salles et pour le répertoire de l'Office des constructions fédérales. L'idée n'est donc pas nouvelle. La base de données actuelle (Oracle 8i) est stockée sur une machine HP AlphaServer DS20E sous Unix, tandis que l'applicatif repose sur la suite Oracle Internet Application Server (serveur Web Apache), et est hébergé sur une machine biprocesseur sous Windows 2000. Le tout est utilisable depuis n'importe quel point du globe sans installation propriétaire spécifique puisque les logiciels clients sont des browsers Web. Dès lors, toutes les données Locaux2 sont-elles publiques ? Non, bien sûr. Seuls les utilisateurs référencés peuvent y avoir accès, via les transactions sécurisées du i-moniteur (<http://infowww.epfl.ch/Imon-public/menu.htm>), que ce soit en mode administrateur (service des bâtiments), gestionnaire (responsables des locaux dans les facultés) ou lecteur (divers services centraux). Mais parmi les centaines de données gérées dans la trentaine de tables Oracle, certaines données choisies sont mises à disposition d'applications telles que le bottin ou l'outil d'orientation géographique et sont donc ainsi rendues publiques (notez que les accès gestionnaire et lecteur ne sont pas encore ouverts à ce jour, mais cela ne saurait tarder).

Justement, quelles sont les données gérées? Elles sont trop nombreuses pour être citées en détail ici, mais on peut relever: les codes analytiques et statistiques pour la Conférence universitaire suisse (CUS et SIUS), les types de sols, de faux-plafonds, d'éclairage naturel, de places potentielles, d'utilisation, d'occupation, d'espace CEPF et DIN 277, de même que le rattachement aux centres financiers, les différentes surfaces et volumes, la journalisation des mutations, les dates de validité, les zones, et même des photos de certains bâtiments et espaces.

Pour mesurer les surfaces de tous ces espaces, pas besoin d'user 3 doubles-mètres, même si quelqu'un a bien dû s'y mettre une fois ;-). En fait, les surfaces sont automatiquement reprises des plans AutoCAD 2000 du SBAT, via le module Espace du progiciel de Facility Management Archibus-AFM; ce dernier (qui est notamment utilisé pour effectuer des impressions professionnelles des plans) reçoit cependant une compensation: des données de Locaux2, qu'il peut dès lors utiliser pour ses impressions. A court terme, ce sont ces mêmes données qui seront mises à disposition de l'outil d'orientation géographique de manière totalement automatique.

On commence donc à entrevoir la rationalisation, l'utilité et l'efficacité induites par l'imbrication de toutes ces bases de données et on peut dès lors en comprendre les enjeux: une modification à un endroit est très rapidement répercutée et utilisée en bien d'autres. Et les bénéficiaires en sont très divers: les services centraux, qui peuvent fournir de

meilleurs services dans des délais plus brefs donc moins chers; les visiteurs (de même que les étudiants et les employés), qui se retrouvent plus facilement sur le campus; et dans quelques semaines, les facultés, qui n'auront pas besoin de réinventer la roue, puisque des données de grande qualité leur seront mises à disposition sans aucun frais.

Pour conclure, permettez-moi de remercier pour leur excellente collaboration: e-pfl et CampToCamp pour l'outil d'orientation géographique, DBAssociates pour AFM, et bien entendu le Service des bâtiments. La mise en commun de nos ressources et connaissances métier spécifiques s'est révélée fructueuse et efficace. ■

## SUITE DE LA PREMIÈRE PAGE

### Services Web & Web des services

comme *naïve Bayes*, *arbres de décision*, *réseaux de neurones (PMC)*, *machines à vecteurs supports (SVM)*, *modèles de Markov cachés (MMC)*, *apprentissage relationnel*. Notre service fait appel à un classifieur SVM. Expliqué de façon simple, imaginez un tas de ronds et de losanges dans un espace à 2 dimensions. Le classifieur va séparer au mieux ces deux tas. Imaginons que seuls les ronds vous intéressent, étant donné un candidat carré, le classifieur va apprécier la distance qui sépare ce candidat des ronds. Avec des documents, on fait la même chose en les plongeant dans un espace de grande dimension (sous-ensemble des termes quelque peu modifiés des documents). A titre indicatif voici quelques-uns des sites qui sont visités plusieurs fois par jour, depuis quelques mois. Toutes les pages visitées sont gardées en mémoire ce qui permet d'identifier les changements d'un passage à l'autre:

- [www.aldaily.com/](http://www.aldaily.com/)
- [www.scitechdaily.com/](http://www.scitechdaily.com/)
- [abc.net.au/science/news/default.htm](http://abc.net.au/science/news/default.htm)
- [www.nature.com/nsu/](http://www.nature.com/nsu/)
- [www.nature.com/nature/insights/index.html](http://www.nature.com/nature/insights/index.html)
- [www.scicentral.com/](http://www.scicentral.com/)
- [www.sciencedaily.com/news/headlines.htm](http://www.sciencedaily.com/news/headlines.htm)
- [www.nas.edu/headlines](http://www.nas.edu/headlines)
- [www.nationalacademies.org/topnews/](http://www.nationalacademies.org/topnews/)
- [www.sciencenews.org/index.asp](http://www.sciencenews.org/index.asp)
- [exn.ca/ScienceNews/index.asp](http://exn.ca/ScienceNews/index.asp)
- [www.sciam.com/news\\_directory.cfm](http://www.sciam.com/news_directory.cfm)
- [www.space.com/news/index.html](http://www.space.com/news/index.html)
- [www.wired.com/news/nc\\_index.html](http://www.wired.com/news/nc_index.html)
- [worldscientist.com/](http://worldscientist.com/)
- [www.closetotruth.com](http://www.closetotruth.com)

### LE CLIENT SOAP

SOAP est un protocole de transmission de messages XML. On peut l'utiliser dans le cadre de simples transmissions unidirectionnelles comme dans celui de dialogues de type requête-réponse. Bien que mettant en avant HTTP, SOAP n'est pas lié à un protocole de transport particulier. Il est également indépendant du système d'exploitation et du langage de programmation.

La syntaxe d'un message SOAP est décrite en détail dans un document du W3C ([www.w3.org/TR/SOAP/](http://www.w3.org/TR/SOAP/)).

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://www.w3c.org/..."
xmlns:xsi="http://schemas.xmlsoap.org/soap/encoding/">
<SOAP-ENV:Header></SOAP-ENV:Header>
<SOAP-ENV:Body>
<ns1:GetNews
xmlns:ns1="urn:BasicFilteringService">
<param1 xsi:type="xsd:int">123</param1>
</ns1:uneMethode>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Un message SOAP est constitué d'une enveloppe contenant un en-tête (*header*) facultatif et le corps du message (*body*). La plupart des packages SOAP que vous utiliserez prendront en charge les détails de syntaxe des messages SOAP, mais pour se fixer les idées, regardons un message de requête dans le cadre d'un appel de service:

Le prologue XML contient seulement une déclaration XML `<?xml version="1.0" encoding="UTF-8" ?>` spécifiant la version de XML et l'encodage des caractères du message XML.

L'enveloppe SOAP est marquée par la balise `<SOAP-ENV:Envelope ... >`. Elle spécifie que le style d'encodage de ce message SOAP suit le schéma défini dans <http://schemas.xmlsoap.org/soap/encoding/>. L'élément en-tête est optionnel. Vient ensuite le corps désigné par la balise `<SOAP-ENV:Body>` qui contient la méthode à appeler `<ns1:getNews... >` accompagné de son namespace `urn:BasicFilteringService`. Cette balise encapsule à son tour un paramètre `<param1 ... >`. Les paramètres n'ont pas de namespace.



figure 1



Vous ne pouvez pas utiliser un service sans avoir son descripteur de déploiement. J'ai choisi d'enregistrer le service dans un environnement Apache SOAP. Apache SOAP (<http://xml.apache.org/soap/index.html>) est une implémentation d'un service SOAP en Java de l'Apache Software Foundation. La figure 1 vous donne le résultat de ce déploiement. Vous y trouvez son nom, plus précisément l'URN du service *urn:BasicFilteringService*, le nom du package et les méthodes qui sont au nombre de trois:

- *GetCategories* pour obtenir la liste des catégories.
- *GetNews* pour obtenir les nouvelles des douze dernières heures pour une catégorie ou pour l'ensemble.
- *ExecuteQuery* pour faire une recherche dans une catégorie ou sur l'ensemble.

Nous avons toutes les informations nécessaires pour nous lancer dans l'écriture de notre premier client SOAP (*SoapClient1.java*). Le but fixé est la recherche des nouvelles du jour dans la catégorie physics. Le programme commence par créer une instance de `java.net.URL` sur le service de routage *rpcrouter* (<http://soap1.epfl.ch:8080/soap> est l'URL du service SOAP sur *Tomcat*). La méthode *setTargetObjectURI()* appliquée à un objet *org.apache.soap.rpc.Call* fixe l'URI du service et *setMethodName()* le nom de la méthode qui à la lumière du descripteur est *getNews*:

```
URL url = new URL("http://soap1.epfl.ch:8080/
                 soap/servlet/rpcrouter");
Call call = new Call();
call.setTargetObjectURI
    ("urn:BasicFilteringService");
call.setMethodName("getNews");
```

Le code concernant le passage, plus exactement la sérialisation des paramètres à envoyer se fait de la façon suivante:

```
call.setEncodingStyleURI(Constants.NS_URI_SOAP_ENC);
Object[] multiParams = { categorie };
Vector params = new Vector();
params.addElement(new Parameter("params",
    Object[].class, multiParams, null));
call.setParams(params);
```

On commence par préciser le style d'encodage (ici *Constants.NS\_URI\_SOAP\_ENC* équivalent à <http://schemas.xmlsoap.org/soap/encoding>), puis on range les paramètres (dans cet exemple un seul, le nom de la catégorie) dans un vecteur *params* et on les associe à l'objet *call*.

En invoquant la méthode *invoke()* sur l'objet *call* on capture la réponse du service. Sur cette instance *Response* on applique la méthode *getReturnValue()* pour obtenir une instance de *Parameter*. On récupère la valeur du paramètre de retour par *getValue()*:

```
try {
    call.setMethodName("getNews");
    Response resp = call.invoke(url, "");
    Parameter ret = resp.getReturnValue();
    Object value = ret.getValue();
    System.out.println("Réponses: " + value);
} catch (SOAPException e) {
    System.err.println("Caught SOAPException (" +
        e.getFaultCode() + "): " +
        e.getMessage());
}
```

En ajoutant les imports

```
import java.net.*;
import java.util.*;
import org.apache.soap.*;
import org.apache.soap.rpc.*;
```

et le morceau de code pour saisir la catégorie, vous pouvez passer à la compilation et au test:

```
java SoapClient1 physics
```

```
Réponses: [physics]<a href="http://www.first
science.com/site/articles/blackholes.asp">First
Science.com - Black Holes and Time Machines</a>
```

La réponse se présente sous la forme suivante: le nom de la catégorie entre crochets, une suite de liens (au format HTML) retenus par l'automate qui sont identifiés par leur titre.

Modifions légèrement *SoapClient1.java* pour exploiter la méthode *executeQuery*. Dans ce cas nous devons sérialiser deux paramètres qui sont le nom de la catégorie et la chaîne de recherche:

```
Object[] multiParams = { name, request };
```

et invoquer un nouveau service *executeQuery*:

```
call.setMethodName("executeQuery");
```

La requête *Quelles sont les références à des articles retenus par l'automate sur l'ensemble des catégories où les mots human et francis sont présents?* s'exprime de la manière suivante:

```
java SoapClient2 "human francis"
```

```
Réponses: [biology]<a href="http://www.nhgri.
nih.gov/NEWS/sequencing.html">NHGRI Prioritizes
Next Organisms to Sequence</a>[biology]<a href=
"http://www.sciencedaily.com/releases/2002/05/
020508073140.htm">ScienceDaily Magazine -
International Team Of Researchers Assembles
Draft Sequence Of Mouse Genome</a>[biology]
<a href="http://www.newscientist.com/news/news.
jsp?id=ns99992253">New Scientist</a>
[consciousness]<a href="http://cogprints.soton.
ac.uk/view-evol-psy.html">Cogprints - Subject:
Evolutionary Psychology</a>.....
```

À la date d'aujourd'hui il existe une bonne douzaine d'implémentations de SOAP. La question de l'interopérabilité est ouverte. Comment un client Apache SOAP discute-t-il avec un service sur .NET ou GLUE ? Il existe une tentative de proposer une grammaire XML pour décrire ces services Web. Cette proposition soutenue par Ariba, IBM et Microsoft auprès du W3C s'appelle WSDL (Web Services Description Language). WSDL va dans la bonne direction, mais la génération des URIs qui ne sont pas globalement uniques dans le temps indique que le travail sur WSDL n'est pas tout à fait abouti.

Signalons le **Java Web Services Developer Pack** (WSDP) qui va dans cette direction, il est téléchargeable (32 Mo) directement depuis le site de Sun. Il contient un ensemble d'APIs et d'outils qui étendent la plate-forme Java 2 pour lui permettre de créer, de déployer et d'exécuter des services Web conformes aux spécifications SOAP, WSDL et UDDI.

L'image suivante est tirée de la collection O'Reilly: Java & XML, elle résume bien l'esprit d'un service Web dans la culture SOAP.

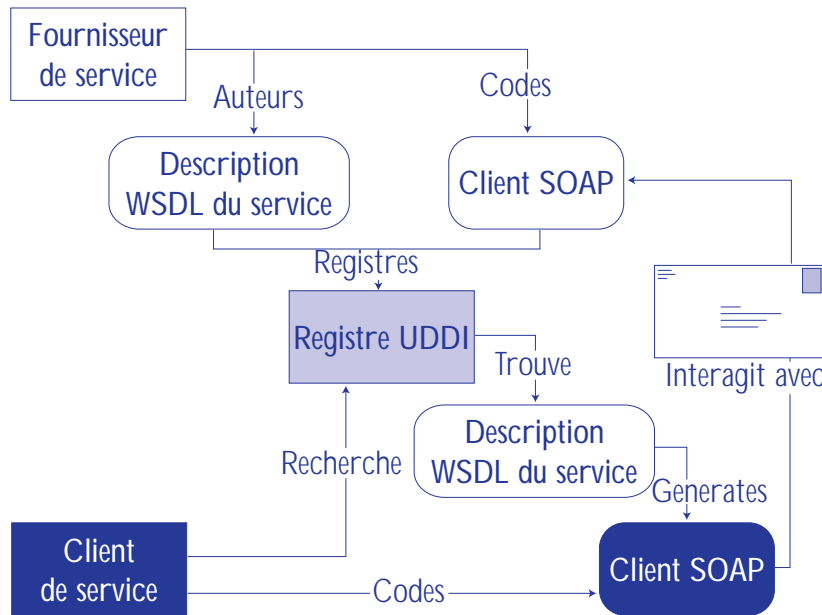


figure 2

Sans aller plus loin avec SOAP, passons au réseau pair-à-pair.

## P2P

### introduction

Un constat s'impose. L'exploitation de ressources massivement distribuées est aujourd'hui une préoccupation majeure qui va au-delà des services Web. Suivant les centres d'intérêts on distingue plusieurs approches qui vont toutes dans ce sens:

- Calcul global (Global Computing) qui consiste à récupérer, à l'image du projet SETI@home, RSA-155 ou Decrypton, des cycles de nœuds de calcul distribués sur le réseau.

- Calcul sur grille (Grid Computing) qui consiste à conduire, à l'image de Globus ([www.globus.org](http://www.globus.org)), de très gros calcul de modélisation ou de simulation en utilisant les ressources de grands centres de calcul distribués dans le monde.
- Calcul pair-à-pair (Peer-to-Peer, P2P) qui consiste à établir une interconnexion d'égal à égal ou de pair à pair, entre plusieurs ordinateurs. Il complète le modèle classique client/serveur, en symétrisant la relation des nœuds qui interagissent. La relation client-serveur n'est plus associée aux nœuds mais aux transactions: chaque nœud peut être client dans une transaction et serveur dans une autre (figure 3).

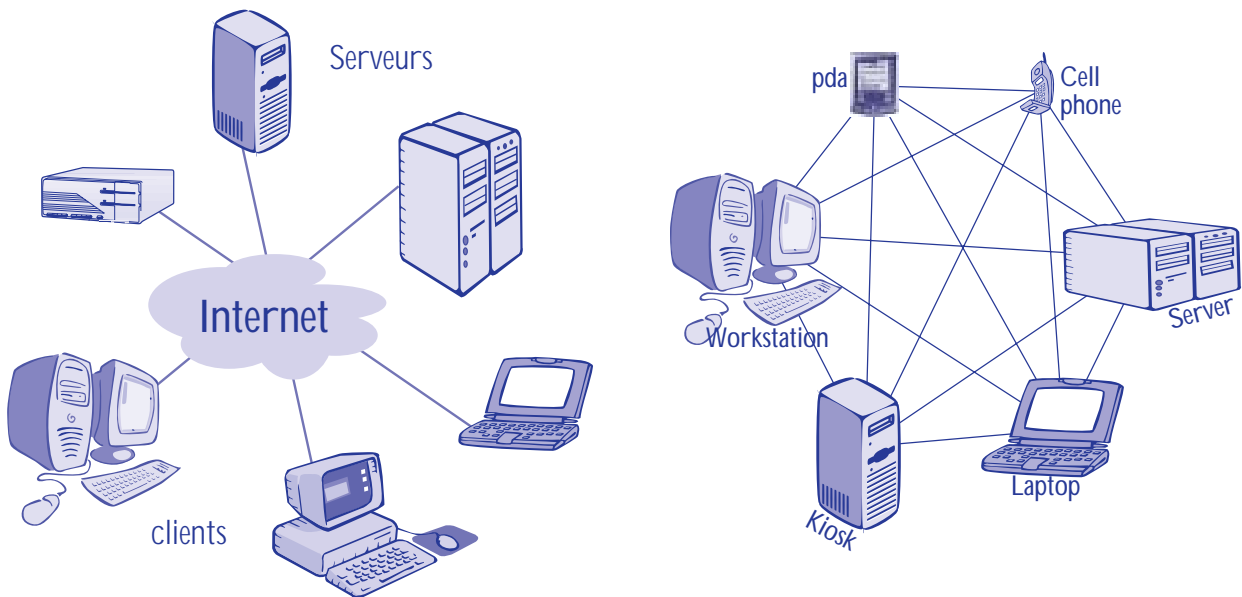


Figure 3

Ce constat fait, nous allons, à l'image de l'exercice précédent avec SOAP, choisir une implémentation dans le cadre d'un réseau pair-à-pair. Le choix s'est porté sur JXTA. Nous allons commencer par introduire les mécanismes de base de JXTA puis passer au codage du client du service *BasicFilteringService*.

Pour celles et ceux qui voudraient aller plus loin avec JXTA, je vous conseille l'excellent livre de Brendon Wilson dont vous pouvez télécharger gratuitement le fichier PDF du Web ou bien l'acheter sous forme papier (<http://www.brendonwilson.com>).

## JXTA, une infrastructure pour des services pair-à-pair

Le projet JXTA (<http://www.jxta.org>) propose des concepts et mécanismes de base génériques propres aux réseaux pair-à-pair. Initié par Sun dans une approche Open Source ce projet identifie plusieurs protocoles permettant de gérer un certain nombre de fonctionnalités communes comme la gestion des pairs, la découverte des ressources, l'invocation de services, la communication inter-pairs, la sécurité, etc. Ces protocoles servent de briques de base pour la mise en œuvre de services pair-à-pair.

## CONCEPTS DE BASE

### Pair

Un pair est l'unité structurelle de base de tout réseau P2P. Un pair peut être toute ressource capable de se connecter au réseau et de fournir un service: poste de travail, grappe de machines ou même un assistant personnel électronique de type PDA. JXTA distingue plusieurs types de pairs:

#### Pairs minimaux

Ces pairs sont réduits à des fonctionnalités élémentaires de communication: envoi et réception de messages. Ils ne sont capables ni de stocker de l'information ni de *router* les messages.

#### Pairs simples

Ces pairs peuvent envoyer et recevoir des messages et en plus stocker dans un cache local des informations sur les ressources du réseau sous forme d'annonces. Ces annonces leur permettent de répondre à des requêtes d'informations et de participer ainsi au protocole de découverte de ressources.

#### Pairs de rendez-vous

Ces pairs sont essentiels à la constitution du réseau. En plus des fonctionnalités décrites plus haut ces pairs permettent de faire suivre les requêtes vers d'autres pairs connus, pairs simples ou pairs de rendez-vous.

#### Pairs de routage

Ces pairs stockent des informations de routage vers les autres pairs. En plus ils permettent de transmettre les requêtes des pairs isolés du réseau par des mécanismes de pare-feu.

#### Groupe de pairs

Les groupes de pairs correspondent à des ensembles de pairs réunis par un intérêt commun: une application de collaboration, un ensemble de services ou un niveau de sécurité.

## Service

Les services sont des traitements fournis au sein d'un réseau P2P (recherche, partage, etc.). Ce sont eux qui motivent la constitution du réseau. On distingue deux types de services, des services individuels fournis par des pairs individuels et des services de groupe fournis par un groupe à tous ses membres. Plusieurs membres du groupe peuvent contribuer, de manière redondante, à ce dernier type de service qui reste disponible tant qu'au moins un membre du groupe reste connecté.

## Annonce

Une annonce est une représentation structurée d'une entité présente sur le réseau. En particulier, toutes les entités décrites jusqu'ici (pair, groupe de pairs, service) ou toute autre ressource peut être décrite par une annonce.

## Protocoles de base

Plusieurs protocoles définissent la structure des échanges d'informations pour un ensemble d'interactions fréquentes dans un réseau P2P. JXTA définit 6 protocoles que nous décrivons dans la section suivante.

### Protocoles JXTA

Les protocoles JXTA sont définis sous forme de séries de messages XML et correspondent aux principaux types d'interaction présents dans un réseau P2P. Tous les protocoles sont asynchrones et reposent sur un modèle requête/réponse. Typiquement un pair envoie une requête à un ou plusieurs pairs de son groupe et peut recevoir zéro, une ou plusieurs réponses. Il n'est pas requis que tous les pairs implémentent tous les protocoles.

#### Protocole de découverte

Ce protocole PDP (Peer Discovery Protocol) est utilisé par les pairs pour faire connaître leurs propres ressources et pour découvrir les ressources fournies par les autres pairs.

#### Protocole d'information

Ce protocole PIP (Peer Information Protocol) permet d'obtenir des informations sur les pairs (état, trafic, etc.).

#### Protocole d'invocation de service

Ce protocole PRP (Peer Resolver Protocol) permet de transmettre des requêtes génériques sur le réseau. Les requêtes peuvent être adressées à un pair spécifique ou bien peuvent être propagées à tous les membres d'un groupe via les pairs de rendez-vous.

#### Protocole de communication par canaux

Le protocole PBP (Pipe Binding Protocol) permet d'établir des canaux virtuels de communication pipes entre un ou plusieurs pairs.

#### Protocole de routage

Ce protocole ERP (Endpoint Routing Protocol) de plus bas niveau est utilisé par les pairs pour déterminer la route d'accès à une destination donnée.

## Protocole de rendez-vous

Le protocole RVP (Rendez-vous Protocol) sert à propager des messages au sein d'un groupe de pairs. Il permet aux pairs de se connecter au service et contrôle la propagation des messages.

Ces six protocoles offrent une couche *middleware* pour l'implémentation de services P2P.

## CODAGE DU CLIENT

Qu'il s'agisse de SOAP ou de JXTA, le client doit s'attacher un service, formuler une requête sous la forme d'un message ou d'une annonce XML, attendre la réponse. A la lumière de ce qui précède le protocole PRP d'invocation de service répond exactement à notre besoin. Le protocole PRP prend en charge l'envoi par un pair d'une annonce de type Resolver Query Message à un *handler* qui se trouve quelque part sur le réseau. Ce *handler* va traiter cette requête et renvoyer une réponse de type Resolver Response Message.

L'interface QueryHandler est le mécanisme proposé par JXTA pour écrire son propre *handler*. Comme ceci concerne le service, nous ne le détaillerons pas explicitement ici mais sachez que cela se résume à écrire la classe

```
class BasicFilteringService implements
    QueryHandler
```

J'ai extrait 3 lignes de ce code pour illustrer les trois étapes du service:

```
// Parse the message from the query string.
eq = new MyQueryMsg(new ByteArrayInputStream(
    (query.getQuery()).getBytes());
// Perform the Query
    answer = executeQuery(eq.getCat(),
        eq.getReq());

// Create the response message.
MyResponseMsg er = new MyResponseMsg(answer);
response = new ResolverResponse
    ("BasicFilteringService",
    null, query.getQueryId(), er.toString());

return response;
}
```

Le corps du client se ramène à la création d'une annonce de type Resolver Query Message,

```
MyQueryMsg equery = new MyQueryMsg (categorie,
    requete);
```

de récupérer l'identifiant du pair,

```
String localPeerId = currentGroup.getPeerID().
    toString();
```

d'invoquer le service

```
ResolverQuery query = new ResolverQuery("Basic
    FilteringService",
    null, localPeerId, equery.toString(), 0);
```

toString() est une méthode de la classe TheQueryMsg

```
public String toString()
{
    try
    {
        StringWriter out = new StringWriter();
```

```
StructuredTextDocument doc =
    (StructuredTextDocument)
    getDocument(new MimeMediaType("text/xml"));
doc.sendToWriter(out);

return out.toString();
}
catch (Exception e)
{
    return "";
}
}
```

et d'envoyer l'annonce à l'ensemble des pairs:

```
resolver.sendQuery(null, query);
```

Comment tester tout ceci ? Installer JXTA sur votre machine (ça prend 1 minute). Lancer le Shell JXTA:

- sous Windows Start->Programs->JXTA->JXTAShell,
- sous linux allez dans le répertoire *shell* et exécutez *run.sh*.

Si tout est en ordre une fenêtre du genre de celle de la figure 4 apparaît. Donnez un nom à votre pair, faites ok et rentrez un *user name* et *password*. Une autre fenêtre apparaît, c'est celle du Shell JXTA (figure 4), lancer la commande *exit*.

Téléchargez l'exemple *jxta\_demo.jar* (<http://...>), installez les classes à l'endroit où vous avez lancé la commande *run.sh* et n'oubliez pas de modifier le CLASSPATH de *run.sh* en conséquence. Vous venez d'ajouter une nouvelle commande au Shell JXTA intitulé *ClientJXTA*. Lancer de nou-

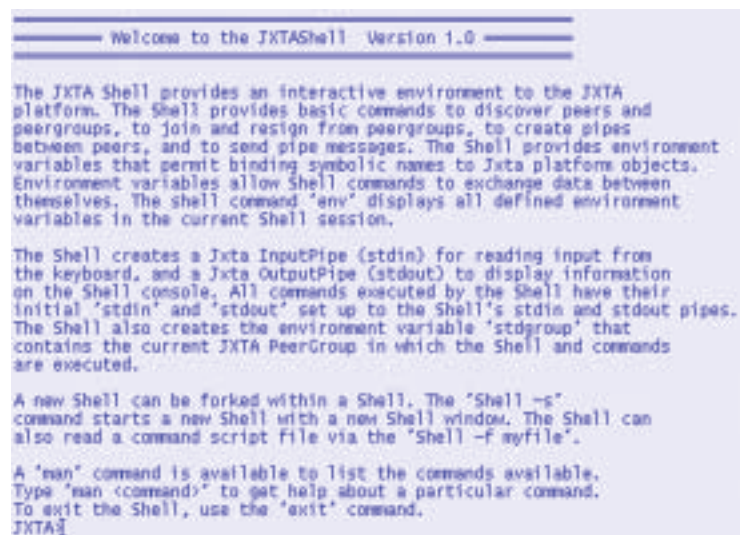


figure 4

veau *run.sh*, *user* et *password* rentrés auparavant , passez la commande *man* , la nouvelle commande *ClientJXTA* doit être dans la liste des commandes. Si ce n'est pas le cas vous avez un problème avec votre CLASSPATH. Si c'est bon lancer la commande *ClientJXTA -c biology -q adn* .

Cette commande va lancer une requête pour le service *BasicFilteringService* qui se trouve quelque part sur le réseau P2P (sur le pair *f101*) et renvoyer dans une fenêtre console une réponse:

```
Processing response...
Searched the category biology for adn:
[biology]<a href="http://human-nature.com/nibbs/02/caldararo.html">Ancient DNA and Human origins: The role of gene sequence variation in the species concept by Niccolo Caldararo</a>
```

Si vous n'avez aucune réponse, vérifiez ma présence sur le réseau P2P (pair *f101*), avec la commande *peers*. En cas d'absence faites moi parvenir un mail pour que j'inscrive le service (fig. 4).

Si vous comparez la figure 5 tirée de l'ouvrage de Brendon Wilson, à celle du monde SOAP vous avez une bonne idée des approches différentes d'un *peer* et SOAP services.

## Jahia

Jahia est un serveur d'applications pour la gestion de contenu Web (CMS). En termes moins techniques c'est un environnement de développement collaboratif pour créer et gérer des sites dynamiques. Jahia est un *framework* 100% Java sur une plate-forme J2EE (figure 6) . Les applicatifs Portal Server, CMS et Admin Center s'appuient sur un ensemble de spécifications (méthodes et données) de la couche Jahia Foundation Server.

Le Portal Server prend en charge l'intégration d'applications et services Web , le CMS grâce à l'utilisation de composants de présentation (templates) facilite la création et la mise à jour de sites et l'Admin Center offre un certain nombre d'outils de gestion (utilisateurs, templates, applications).

L'écriture du service se résume à celle d'une *servlet* dans laquelle on définit par la méthode *doGet()*.

```
public class UnService extends
HttpServlet
{
    public void doGet(HttpServletRequest
req, HttpServletResponse res)
throws ServletException, IOException
{
    .....
}
}
```

En tant qu'administrateur d'un site Jahia, vous pouvez enregistrer un nouveau service ou composant au moyen du menu *Manage components* (figure 7). C'est ce que j'ai fait pour notre service *BasicFilteringService* dans le cadre du site *democms.epfl.ch/test*.

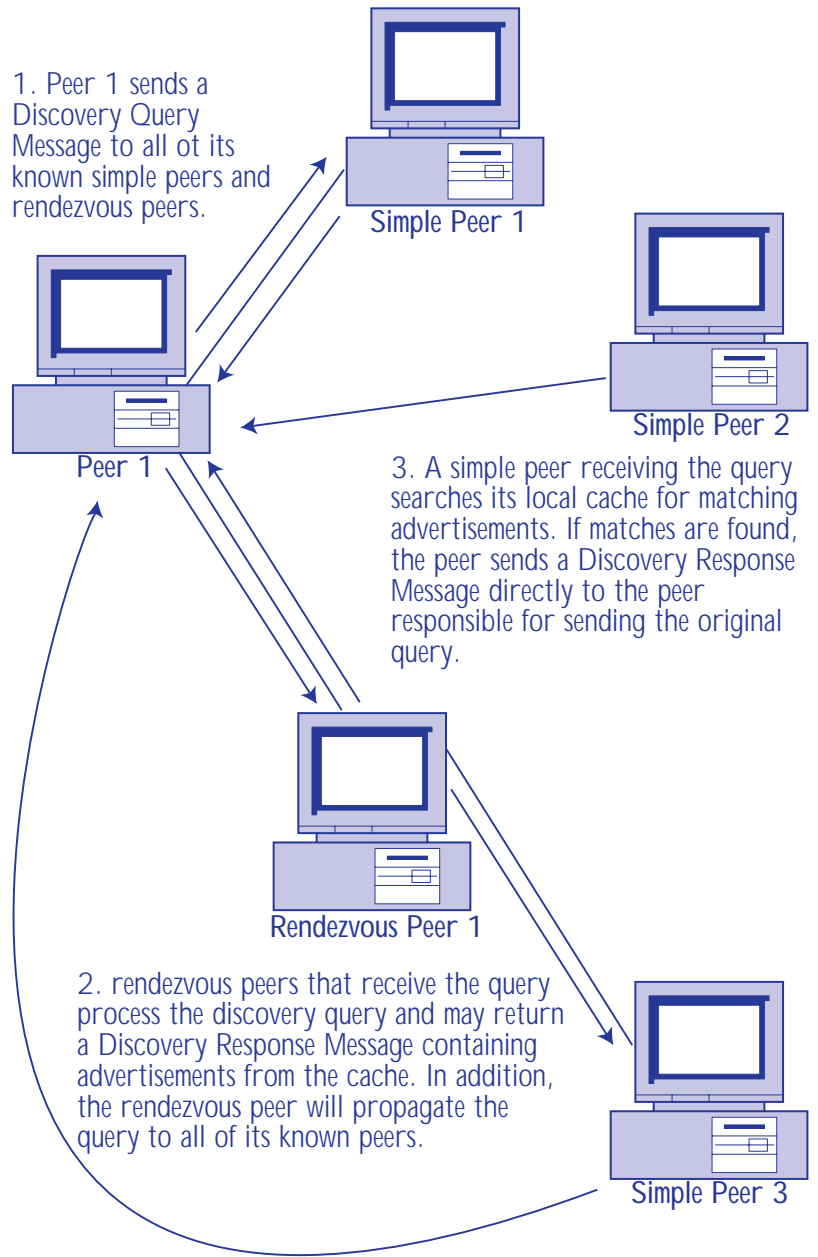


figure 5

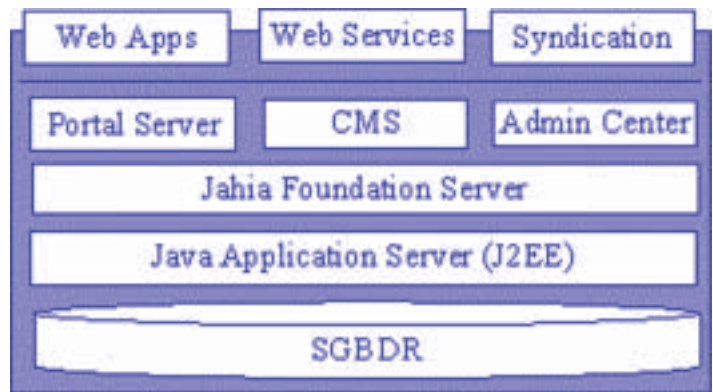


figure 6



figure 7

Voilà pour le côté serveur. Côté client, imaginons que vous êtes éditeur d'un ensemble de pages et que vous voulez vous attacher les services de *BasicFilteringService*. Pour se faire sélectionnez le mode *Edit* dans la barre d'outil. Cliquez *Ajouter une boîte* qui doit contenir le service. Donnez lui un nom *Filtering* et choisissez le type *Webapps* (figure 8).

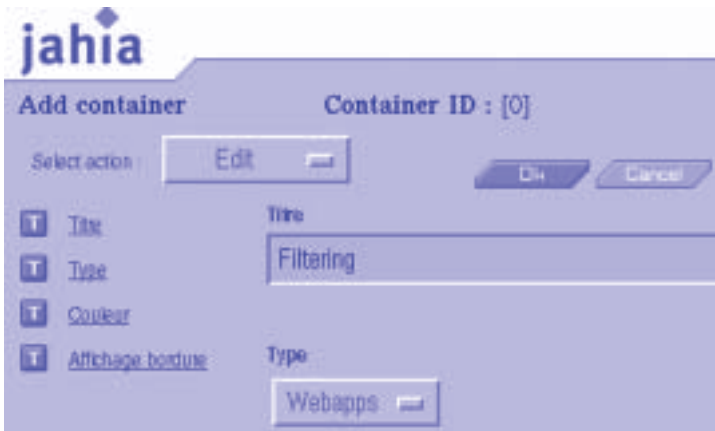


figure 8

En cliquant OK vous devez trouver dans la page la boîte ci-contre:



Cliquez le signe + pour ajouter le service (figure 9):



figure 9

Choisir le service dans la liste qui vous est proposée. Si le service offre des rôles (administrateur, guest,...) cliquez le bouton *Manage* pour l'attribution de ceux-ci. Quitter le menu, et normalement des nouvelles du jour dans la catégorie *Génétique* doivent s'afficher:



(résultat du 18/09/02 après-midi)

## Conclusion

Le but de cet article, au-delà de son aspect didactique, est de souligner l'urgence d'organiser tous les projets d'intérêt général autour de cette idée de mutualisation de services. Suivant notre capacité de réaction trois arithmétique nous attendent:

- 1+1=1 (le tout vaut moins que la somme des parties: **régression**)
- 1+1=2 (**état stationnaire**)
- 1+1=3 (le tout vaut plus que la somme des parties: **évolution**) ■



Renseignements  
(les matins des lu, me & ve)  
Daniele.Gonzalez@epfl.ch  
☎ 021/69 353 14  
Fax: 021/69 322 20

# PROGRAMME DES COURS

*organisés par le Service informatique central de l'EPFL*

Ces cours sont ouverts à tous, membres ou non de l'EPFL.  
Pour le personnel de l'EPFL, le SIC se charge des frais de cours.  
Les descriptifs des cours sont sur Internet: <http://sic.epfl.ch/formation>



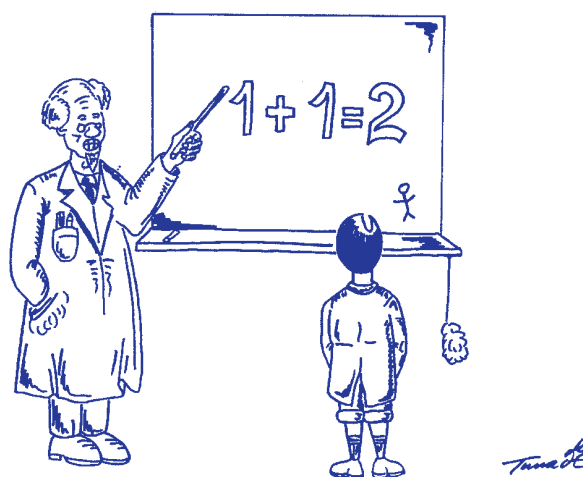
Renseignements  
(tous les matins)  
Josiane.Scalfio@epfl.ch  
☎ 021/69 322 44  
Fax: 021/69 322 20

## CONDITIONS D'INSCRIPTION

*En cas d'empêchement à suivre le(s) cours, l'élève avertira le Service informatique central au minimum une semaine à l'avance (sauf cas exceptionnel), faute de quoi le SIC se réserve le droit de facturer à son unité les frais occasionnés pour le cours.*

*Une confirmation parviendra à l'élève environ deux semaines avant le(s) cours. S'il est déjà complet, l'élève sera informé de suite et son nom placé en liste d'attente. Dès qu'un cours identique sera fixé, il recevra un nouveau formulaire d'inscription.*

*Le SIC se réserve le droit d'annuler un cours si le nombre minimum de 4 participants n'est pas atteint ou pour des raisons indépendantes de sa volonté. Aucune compensation ne sera due par le SIC.*



## INTRODUCTION AU POSTE DE TRAVAIL

OS	Nom du cours	N°	1/2 jour(s)	Date(s)	Horaire
Mac	Entourage, messagerie et News	02-0456	1	11.11.2002	08:30 - 12:00
Win	Internet, la navigation	02-0459	1	29.10.2002	13:30 - 17:00
Mac	Internet, la navigation	02-0455	1	08.11.2002	08:30 - 12:00
Mac	Macintosh, introduction	02-0454	1	05.11.2002	08:30 - 12:00
Mac	Macintosh, votre machine en pratique	02-0360	1	28.10.2002	13:30 - 17:00
Mac	Macintosh, votre machine en pratique	02-0457	1	04.12.2002	13:30 - 17:00
Win	Outlook Express, messagerie et News	02-0460	1	10.12.2002	13:30 - 17:00
Win	Windows 2000, introduction	02-0458	1	28.10.2002	08:30 - 12:00
Win	Windows 2000, votre machine en pratique	02-0461	1	16.12.2002	08:30 - 12:00

## ACQUISITION ET TRAITEMENT DE DONNÉES

OS	Nom du cours	N°	1/2 jour(s)	Date(s)	Horaire
Win	LabVIEW Basics 1	02-0256	6	11 au 13.11.2002	08:30 - 17:00
Win	LabVIEW Basics 2	02-0257	4	14 & 15.11.2002	08:30 - 17:00
Win	LabVIEW DAQ	02-0258	4	12 & 13.12.2002	08:30 - 17:00
Win	LabVIEW Vision IMAQ	02-0409	4	09 & 10.12.2002	08:30 - 17:00

## BASE DE DONNÉES

OS	Nom du cours	N°	1/2 jour(s)	Date(s)	Horaire
Win	Access, 1-introduction	02-0476	4	11 & 18.11.2002 et 12 & 19.11.2002	08:30 - 12:00 13:30 - 17:00
Win	Access, 2-avancé	02-0477	4	02, 05, 09 & 12.12.2002	13:30 - 17:00
Win	FileMaker Pro, 1-introduction	02-0432	1	11.11.2002	13:30 - 17:00
<b>NOUVEAU</b>	Mac FileMaker Pro, 2-mise en forme de modèles	02-0433	1	31.10.2002	13:30 - 17:00
<b>NOUVEAU</b>	Mac FileMaker Pro, 3-liste de valeurs et options	02-0434	1	07.11.2002	13:30 - 17:00
<b>NOUVEAU</b>	Mac FileMaker Pro, 4-création d'une base de données	02-0435	3	20, 21 & 27.11.2002	08:30 - 12:00
Win	FileMaker Pro, atelier d'exercices	02-0437	1	25.11.2002	08:30 - 12:00
Mac	FileMaker Pro, scripts et boutons	02-0436	1	02.12.2002	13:30 - 17:00

## ÉDITION

OS	Nom du cours	N°	1/2 jour(s)	Date(s)	Horaire
Mac	Acrobat (PDF)	02-0452	1	04.11.2002	08:30 - 12:00
Win	FrameMaker, 1-mise en forme	02-0429	3	04, 05 & 11.12.2002	08:30 - 12:00
Win	FrameMaker, 2-livre et EndNote	02-0430	1	12.12.2002	08:30 - 12:00
Mac	Word, atelier d'exercices	02-0374	1	29.10.2002	13:30 - 17:00
Win	Word, images et colonnes	02-0447	1	26.11.2002	08:30 - 12:00
Win	Word, introduction	02-0440	1	18.11.2002	13:30 - 17:00
Win	Word, longs documents	02-0448	1	29.11.2002	08:30 - 12:00
Win	Word, modèles et publipostage (mailing)	02-0451	1	10.12.2002	08:30 - 12:00
Win	Word, outils	02-0450	1	06.12.2002	08:30 - 12:00
Win	Word, styles	02-0441	1	19.11.2002	08:30 - 12:00
Win	Word, tableaux	02-0449	1	03.12.2002	13:30 - 17:00

## OUTLOOK

OS	Nom du cours	N°	1/2 jour(s)	Date(s)	Horaire
Win	Outlook 2000, atelier d'exercices	02-0465	1	04.12.2002	13:30 - 17:00

## PRÉSENTATION

OS	Nom du cours	N°	1/2 jour(s)	Date(s)	Horaire
Mac	PowerPoint, atelier d'exercices	02-0462	1	6.11.2002	13:30 - 17:00
Win	PowerPoint, introduction	02-0438	1	25.11.2002	13:30 - 17:00
Mac	PowerPoint, les présentations	02-0380	2	29 & 31.10.2002	08:30 - 12:00
Win	PowerPoint, les présentations	02-0439	2	02 & 03.12.2002	08:30 - 12:00



## PROGRAMMATION

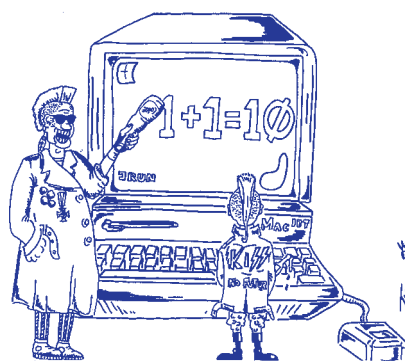
	OS	Nom du cours	N°	1/2 jour(s)	Date(s)	Horaire
<b>NOUVEAU</b>	Win	ASP .NET	02-0327	6	12 au 14.11.2002	08:30 - 17:00
	Win	Java avancé	02-0319	10	18 au 22.11.2002	08:30 - 17:00
	Win	Java Serveurs d'applications J2EE	02-0320	10	25 au 29.11.2002	08:30 - 17:00
	Linux	Langage C++	02-0321	10	09 au 13.12.2002	08:30 - 17:00
<b>NOUVEAU</b>	Win	VB .NET avancé	02-0325	6	28 au 30.10.2002	08:30 - 17:00
<b>NOUVEAU</b>	Win	VB .NET avancé	02-0326	6	05 au 07.11.2002	08:30 - 17:00
<b>NOUVEAU</b>	Win	VB .NET débutant	02-0324	4	23 & 24.10.2002	08:30 - 17:00

## WINDOWS 2000

	OS	Nom du cours	N°	1/2 jour(s)	Date(s)	Horaire
	Win	Windows 2000, administration	02-0488	6	27 au 29.11.2002	08:30 - 17:00
	Win	Windows 2000, configurations clients et serveurs	02-0486	4	20 & 21.11.2002	08:30 - 17:00
	Win	Windows 2000, dépannage	02-0487	2	25.11.2002	08:30 - 17:00
	Win	Windows 2000, performance, tuning	02-0396	2	29.10.2002	08:30 - 17:00
	Win	Windows 2000, prise en charge d'une infrastructure réseau	02-0489	8	02.12.2002 03 au 05.12.2002 & 06.12.2002	13:30 - 17:00 08:30 - 17:00 08:30 - 12:00

## WINDOWS XP Pro

	OS	Nom du cours	N°	1/2 jour(s)	Date(s)	Horaire
<b>NOUVEAU</b>	Win	Windows XP Pro, administrateurs	02-0484	2	06.11.2002	08:30 - 17:00
<b>NOUVEAU</b>	Win	Windows XP Pro, utilisateurs avancés	02-0485	2	07.11.2002	08:30 - 17:00

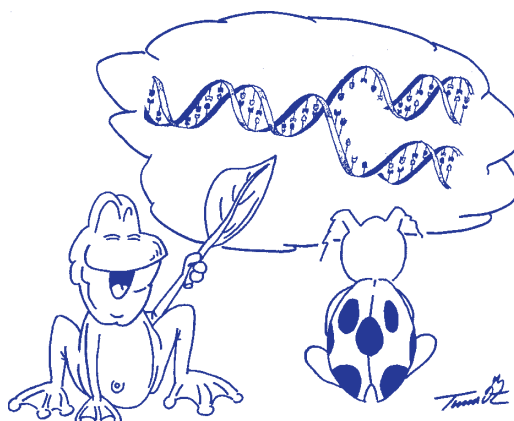


## TABLEUR

	OS	Nom du cours	N°	1/2 jour(s)	Date(s)	Horaire
	Mac	Excel, 1-introduction	02-0413	1	28.10.2002	08:30 - 12:00
	Win	Excel, 1-introduction	02-0453	1	26.11.2002	13:30 - 17:00
	Mac	Excel, 2-feuille de calcul	02-0414	3	11, 13 & 18.11.2002	13:30 - 17:00
	Mac	Excel, base de données	02-0464	2	29.11 & 03.12.2002	08:30 - 12:00
	Mac	Excel, graphiques	02-0415	1	20.11.2002	13:30 - 17:00

## WWW - WEB

OS	Nom du cours	N°	1/2 jour(s)	Date(s)	Horaire
Win	Dreamweaver, 1ère partie	02-0416	2	01.11.2002	08:30 - 17:00
Mac	Dreamweaver, 1ère partie	02-0417	2	18 & 19.11.2002	08:30 - 12:00
Win	Dreamweaver, 2ème partie	02-0466	2	12 & 15.11.2002	08:30 - 12:00
Mac	Dreamweaver, 2ème partie	02-0418	2	25 & 26.11.2002	08:30 - 12:00
Mac	Dreamweaver, atelier d'exercices	02-0426	1	22.11.2002	08:30 - 12:00
Mac	Dreamweaver, avancé	02-0419	2	09 & 10.12.2002	08:30 - 12:00
Mac	Fireworks, création d'éléments graphiques	02-0425	2	09 & 10.12.2002	13:30 - 17:00
Mac	Flash, 1ère partie	02-0423	3	04, 05 & 12.11.2002	13:30 - 17:00
Mac	Flash, 2ème partie	02-0424	2	25 & 26.11.2002	13:30 - 17:00
Mac	GoLive, 1ère partie	02-0420	2	06 & 07.11.2002	08:30 - 12:00
Mac	GoLive, 2ème partie	02-0421	2	13 & 14.11.2002	08:30 - 12:00
Mac	GoLive, atelier d'exercices	02-0463	1	27.11.2002	13:30 - 17:00
Mac	GoLive, avancé	02-0422	2	04 & 05.12.2002	08:30 - 12:00



### INSCRIPTION POUR LES COURS ORGANISÉS PAR LE SIC

*A retourner à Josiane Scalfò ou à Danièle Gonzalez, SIC-EPFL, CP 121, 1015 Lausanne*

Je, soussigné(e) Nom: ..... Prénom: .....

Tél.: ..... E-Mail: ..... Fonction: .....

Institut: ..... Dépt: ..... Adresse: .....

m'engage à suivre le(s) cours dans son (leur) intégralité et à respecter l'horaire selon les conditions d'inscription:

Nom du cours	N° du cours	N° cours de remplacement	Date du cours
.....			

Pour les cours système Windows 2000, choix du support de cours    en français     en anglais

Date: ..... Signature: .....

Autorisation du chef hiérarchique (nom lisible et signature): .....

### INTÉRÊT ET SOUHAIT POUR D'AUTRES COURS

Description ou titre des cours que je souhaite voir organiser par le SIC:

.....

# FLASH SUR FLASH ... DANS LE FLASH !

Hicham Dennaoui, dennaoui@arobasque.ch



Oui, je sais c'est facile ... oui, je sais c'est nul ... oui je sais un article sur Flash dans le Flash ça fait tache. Et pourtant, je vous le demande: A quand le Flash en Flash hein? ... On peut toujours rêver ... Et bien justement, ...Avec Flash, je vous invite à découvrir une nouvelle catégorie d'objets informatiques: les *Dreamware* ...Bienvenue donc, dans le monde merveilleux du rêve. Un monde où tout est possible... un monde sans contraintes ... sans contraintes?!? ... Ah non, pardon ... il faudra vous munir de deux choses extrêmement rares et précieuses: une imagination débordante et surtout du temps. Au pays des *Dreamware*, Flash est roi.

Flash est le programme d'animation de page Web par excellence. On retrouve Flash non seulement dans la bande annonce de certains sites mais également comme éléments esthétiques et dynamiques de l'interface utilisateur de pages Web. Cet article, non content de vous faire rêver, va vous présenter de manière succincte quelques nouveautés de Flash MX, la dernière incarnation de ce *Dreamware*.

Au fur à mesure des nouvelles versions Flash, les utilisateurs de Flash se sont groupés en deux catégories: ceux qui utilisent Flash principalement pour l'animation et le graphisme et ceux qui l'utilisent plutôt pour ses outils de programmation. Flash MX offre des nouveautés intéressantes pour les deux catégories ...

Premier changement qui saute aux yeux ... L'interface utilisateur: les palettes flottantes ont été regroupées pour certaines, ou mieux encore intégrées dans une palette **Propriétés** du même type que celle que les utilisateurs de *Dreamweaver* connaissent bien. La palette Propriétés est une palette *contextuelle* dont le contenu varie selon la sélection de tel ou tel objet. Cliquez sur un symbole et la palette **Propriétés** vous permet de changer le type, la taille, la position dudit symbole.

La *timeline* a elle aussi été améliorée. Il est désormais possible de regrouper les calques dans des dossiers, ce qui facilite grandement la gestion des calques lorsqu'il s'agit de les organiser, de les cacher ou encore de les verrouiller. Et il est plus simple de travailler sur plusieurs images en même temps.

## Tools for fools

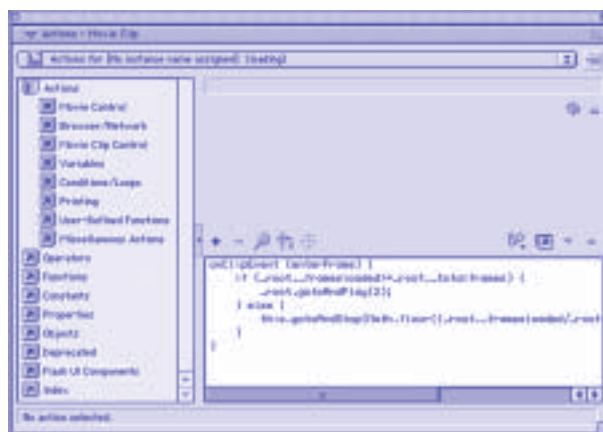
La palette **outils** s'est également vue parée de nouveaux atours pour la plus grande joie des graphistes. L'outil de transformation autorise beaucoup de fantaisie tout en restant très simple à utiliser. La gestion des couleurs bénéficie elle aussi d'un petit *lifting*. Conscient que tous les graphistes n'utilisent pas Flash, Macromedia a amélioré l'intégration de documents Illustrator ou Freehand. Pour les graphistes toujours, un de ces petits détails qui font les grandes différen-

ces: lors de agrandissement à plus de 400%, une grille apparaît sur la scène permettant le positionnement des éléments au pixel près.

Contrairement aux versions précédentes, il est désormais possible d'intégrer différents formats vidéo directement dans votre animation. A l'inverse, si vous voulez des images JPEG ou des sons MP3, vous n'êtes plus obligés de les importer. C'est une amélioration intéressante puisque le changement d'une image ou d'un son ne nécessite plus d'éditer l'animation.



Palette Propriétés



Palette Actions

## PROGRAMMATION

Les **ActionsScripts** permettent de gérer les éléments d'une animation et offrent donc à celui ou à celle que la programmation n'effraie pas une très grande souplesse de manipulation. Flash MX apporte bien sûr son lot de nouvelles actions. Toutefois les améliorations les plus importantes se retrouvent au niveau des outils eux-mêmes: aide en ligne, syntaxe en couleurs, regroupements des **ActionsScripts**, mise en évidence du code, ainsi qu'un nouveau débogueur...

Cette nouvelle mouture de Flash n'est pas une révolution en soi. Mais elle a le mérite d'apporter son lot de petites améliorations et nouvelles fonctionnalités qui rendent la vie plus simple et vous renvoient à cette simple philosophie: avec Flash vos rêves deviennent peut-être des réalités, si ce n'est pas le cas, laissez vos rêves où ils sont et suivez une formation Excel ;-) ■

# CALENDRIER

LU	28.10.02	17 <sup>15</sup>	IN202	Séminaire I&C Info sur: <a href="http://ic.epfl.ch/page6879.html">http://ic.epfl.ch/page6879.html</a>
VE	01.11.02	10 <sup>00</sup>	Salle Conférences SIC	Conférence des Webmasters E. Mc Murray, +41 21 69 35672, courriel: <a href="mailto:Elaine.McMurray@epfl.ch">Elaine.McMurray@epfl.ch</a> Info sur <a href="http://www.myepfl.ch/atelier">http://www.myepfl.ch/atelier</a>
LU	04.11.02	17 <sup>15</sup>	IN202	Séminaire I&C Info sur: <a href="http://ic.epfl.ch/page6879.html">http://ic.epfl.ch/page6879.html</a>
LU	11.11.02	17 <sup>15</sup>	IN202	Séminaire I&C Info sur: <a href="http://ic.epfl.ch/page6879.html">http://ic.epfl.ch/page6879.html</a>
MA	12.11.02	08 <sup>45</sup>	Salle Polyvalente SIC	Comité de rédaction du FI J. Dousson, +41 21 69 32246, courriel: <a href="mailto:Jacqueline.Dousson@epfl.ch">Jacqueline.Dousson@epfl.ch</a>
MA	12.11.02	16 <sup>15</sup>	Salle Wavre	Direction informatique stratégique J.-Cl. Berney, +41 21 69 32590, courriel: <a href="mailto:jean-claude.berney@epfl.ch">jean-claude.berney@epfl.ch</a>
LU	18.11.02	17 <sup>15</sup>	IN202	Séminaire I&C Info sur: <a href="http://ic.epfl.ch/page6879.html">http://ic.epfl.ch/page6879.html</a>
JE	21.11.02	14 <sup>15</sup>	Salle Conférences SIC	PolyPC — Groupe des utilisateurs de PC Ch. Zufferey, +41 21 69 34598, courriel: <a href="mailto:Christian.Zufferey@epfl.ch">Christian.Zufferey@epfl.ch</a> Info sur: <a href="http://pcline.epfl.ch/pc/grp/home.htm">http://pcline.epfl.ch/pc/grp/home.htm</a>
LU	25.11.02	17 <sup>15</sup>	IN202	Séminaire I&C Info sur: <a href="http://ic.epfl.ch/page6879.html">http://ic.epfl.ch/page6879.html</a>
LU	02.12.02	17 <sup>15</sup>	IN202	Séminaire I&C Info sur: <a href="http://ic.epfl.ch/page6879.html">http://ic.epfl.ch/page6879.html</a>
ME	04.12.02	10 <sup>00</sup>	Salle Conférences SIC	Conférence des Webmasters E. Mc Murray, +41 21 69 35672, courriel: <a href="mailto:Elaine.McMurray@epfl.ch">Elaine.McMurray@epfl.ch</a> Info sur <a href="http://www.myepfl.ch/atelier">http://www.myepfl.ch/atelier</a>
LU	09.12.02	17 <sup>15</sup>	IN202	Séminaire I&C Info sur: <a href="http://ic.epfl.ch/page6879.html">http://ic.epfl.ch/page6879.html</a>
MA	10.12.02	08 <sup>45</sup>	Salle Polyvalente SIC	Comité de rédaction du FI J. Dousson, +41 21 69 32246, courriel: <a href="mailto:Jacqueline.Dousson@epfl.ch">Jacqueline.Dousson@epfl.ch</a>
MA	10.12.02	16 <sup>15</sup>	Salle Wavre	Direction informatique stratégique J.-Cl. Berney, +41 21 69 32590, courriel: <a href="mailto:jean-claude.berney@epfl.ch">jean-claude.berney@epfl.ch</a>

