

Autour du web

Une introduction technique

Deuxième partie : XML

Georges-André SILBER

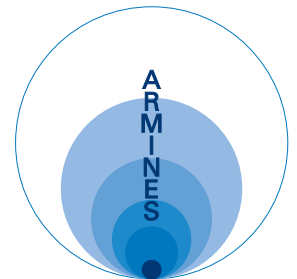
Centre de recherche en informatique

MINES ParisTech

silber@cri.ensmp.fr



<http://www.cri.ensmp.fr/people/silber/cours/2010/web>



Acronymes

- HTML, HTTP, SOAP, WSDL, UDDI, XML-RPC, XML, DTD, XSLT, Schema XML, W3C, DNS, URI, URN, URL, FPI, XSL-FO, XPath, CSS, XHTML, Atom, RSS, Javascript, AJAX, WS-Security, Relax NG, Schematron, DOM, SAX, Podcast, REST, RDF, ...

XML

XML sert au balisage

- De tout temps, l'informaticien s'est intéressé à la représentation des données
- Le fichier est une première révolution importante
- Problème de l'hétérogénéité des fichiers
- Codage ? Structure ? Validité du contenu ?

Georges-André SILBER

Centre de recherche en informatique

Ecole nationale supérieure des mines de Paris

35, rue Saint-Honoré

77305 Fontainebleau cedex

Tél: +33 (0)1 64 69 48 36

Fax: +33 (0)1 64 69 48 47

Por: +33 (0)6 11 11 11 11

Mail: silber@cri.ensmp.fr

Mail: Georges-Andre.Silber@ensmp.fr

Ceci est un fichier: début ? fin ? nom ?
encodage ?

<?xml version="1.0" encoding="utf-8"?>

<adresse>

Georges-André SILBER

Centre de recherche en informatique

Ecole nationale supérieure des mines de Paris

35, rue Saint-Honoré

77305 Fontainebleau cedex

Tél: +33 (0)1 64 69 48 36

Fax: +33 (0)1 64 69 48 47

Por: +33 (0)6 11 11 11 11

Mail: silber@cri.ensmp.fr

Mail: Georges-Andre.Silber@ensmp.fr

</adresse>

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE adresse SYSTEM "adresse.dtd">
<adresse>
<adresse_postale>
<nom>Georges-André SILBER</nom>
<ligne_adresse>Centre de recherche en informatique</ligne_adresse>
<ligne_adresse>Ecole nationale supérieure des mines de Paris</ligne_adresse>
<ligne_adresse>35, rue Saint-Honoré</ligne_adresse>
<codepostal>77305</codepostal>
<ville>Fontainebleau cedex</ville>
</adresse_postale>
<tel type="fixe">+33 (0)1 64 69 48 36</tel>
<tel type="fax">+33 (0)1 64 69 48 47</tel>
<tel type="por">+33 (0)6 11 11 11 11</tel>
<mail>silber@cri.ensmp.fr</mail>
<mail>Georges-Andre.Silber@ensmp.fr</mail>
</adresse>
```

```
<!ELEMENT adresse (adresse_postale,tel+,mail*)>
<!ELEMENT adresse_postale (nom,ligne_adresse+,codepostal,ville)>
<!ELEMENT ligne_adresse (#PCDATA)>
<!ELEMENT codepostal (#PCDATA)>
<!ELEMENT nom (#PCDATA)>
<!ELEMENT ville (#PCDATA)>
<!ELEMENT tel (#PCDATA)>
<!ATTLIST tel
  type (fixe|fax|por) "fixe">
<!ELEMENT mail (#PCDATA)>
```


La norme XML

- XML : eXtensible Markup Language
- Langage de balisage extensible
- Norme du W3C depuis 1998
- Version 1.0 (février 1998)
- Version 1.1 (février 2004)

Origine et buts de XML

- HTML 1.0, 2.0, 3.0, 4.0, 4.1
 - Volonté de “stabiliser” le langage du Web
- Comment : en créant un meilleur HTML
- Inspiration : SGML (Norme ISO 1986)
- Séparation données / présentation
 - Données : XML (ou XHTML)
 - Présentation: CSS, XSLT, etc...



Extensible Markup Language (XML) 1.1

W3C Recommendation 04 February 2004, edited in place 15 April 2004

This version:

<http://www.w3.org/TR/2004/REC-xml11-20040204/>

Latest version:

<http://www.w3.org/TR/xml11>

Previous version:

<http://www.w3.org/TR/2003/PR-xml11-20031105/>

Editors:

Tim Bray, Textuality and Netscape <tbray@textuality.com>

Jean Paoli, Microsoft <jeanpa@microsoft.com>

C. M. Sperberg-McQueen, W3C <cmsmcq@w3.org>

Eve Maler, Sun Microsystems, Inc. <eve.maler@east.sun.com>

François Yergeau <fyergeau@alis.com>

John Cowan <cowan@ccil.org>

Please refer to the [errata](#) for this document, which may include some normative corrections.

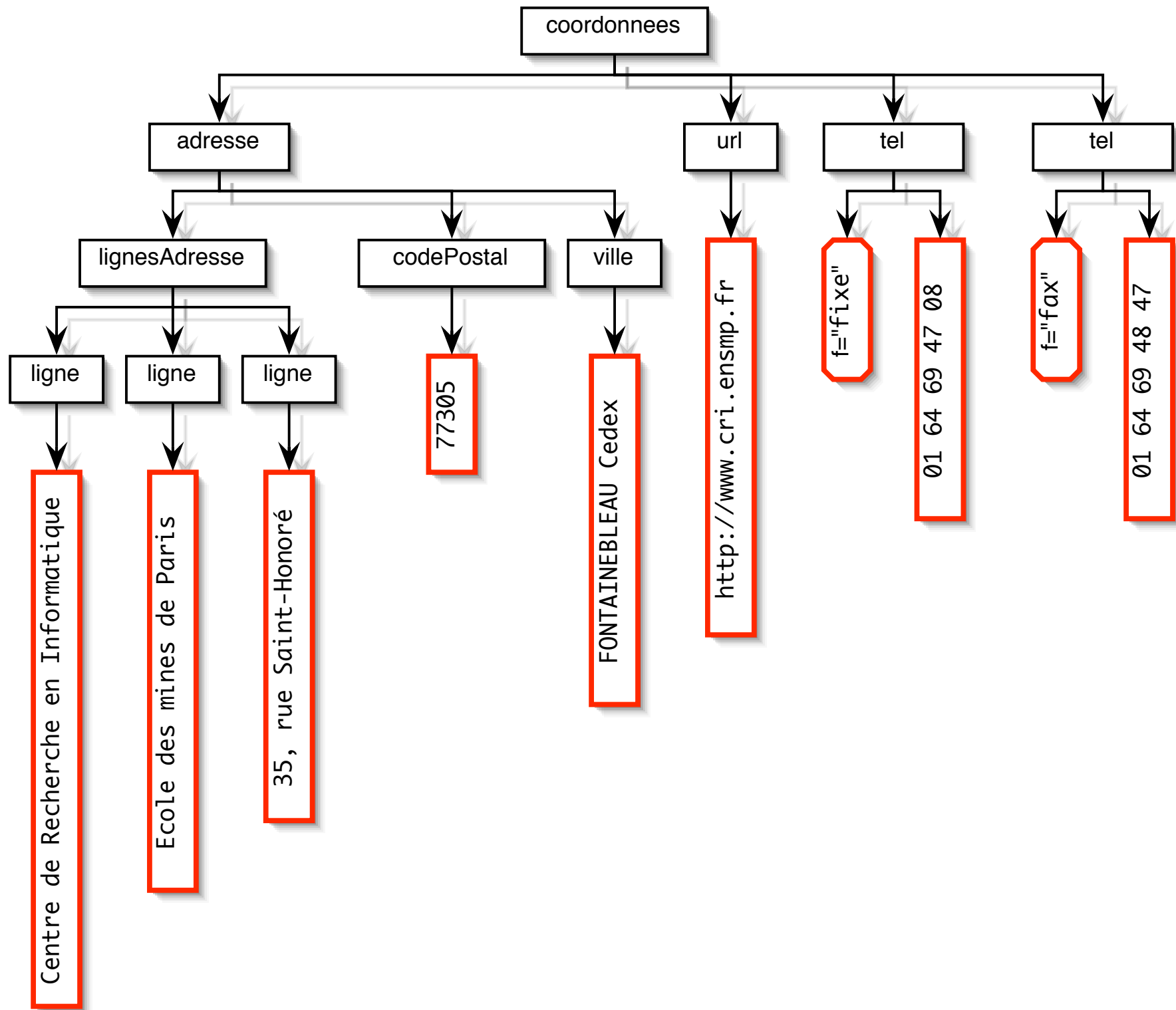
This document is also available in these non-normative formats: [XML](#) and [XHTML with color-coded revision indicators](#)

Qu'est-ce que XML ?

- Un langage de description d'une classe d'objets de données appelés documents XML
- La norme décrit partiellement le comportement de programmes les manipulant
- XML est une forme restreinte de SGML (1986)

Exemple de document XML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<coordonnees>
  <adresse>
    <lignesAdresse>
      <ligne>Centre de Recherche en Informatique</ligne>
      <ligne>Ecole des mines de Paris</ligne>
      <ligne>35, rue Saint-Honoré</ligne>
    </lignesAdresse>
    <codePostal>77305</codePostal>
    <ville>FONTAINEBLEAU Cedex</ville>
  </adresse>
  <url>http://www.cri.ensmp.fr</url>
  <tel t="fixe">01 64 69 47 08</tel>
  <tel t="fax">01 64 69 48 47</tel>
</coordonnees>
```



Documents XML

- Composés d'unités de stockage appelées éléments, contenant des données analysées syntaxiquement (parsées) ou non.
- Ces données sont des caractères qui sont soit des données simples soit des données concourant au marquage (balisage).

Éléments, etc...

```
<?xml version="1.0" encoding="iso-8859-1"?>
```

```
<coordonnees> balise ouvrante
```

```
<adresse>
```

```
<lignesAdresse>
```

```
<ligne>Centre de Recherche en Informatique</ligne>
```

```
<ligne>Ecole des mines de Paris</ligne>
```

```
<ligne>35, rue Saint-Honoré</ligne>
```

```
</lignesAdresse>
```

```
<codePostal>77305</codePostal>
```

élément

```
<ville>FONTAINEBLEAU Cedex</ville> élément
```

```
</adresse>
```

données simples

```
<url>http://www.cri.ensmp.fr</url>
```

```
<tel t="fixe">01 64 69 47 08</tel>
```

```
<tel t="fax">01 64 69 48 47</tel>
```

balise fermante

```
</coordonnees> attribut
```


Documents XML

- Le balisage décrit les structures logiques et de stockage du document
- Un document XML est BIEN FORME il respecte les règles de XML
- XML fournit un mécanisme pour contraindre ces structures, les DTD (Définition de Type de Document)
- Un document XML peut être VALIDE par rapport à une ou plusieurs DTD

Exemple de DTD

```
<!ELEMENT coordonnees (adresse, url?, tel*)>  
<!ELEMENT adresse (lignesAdresse, codePostal, ville)>  
<!ELEMENT lignesAdresse (ligne+)>  
<!ELEMENT ligne (#PCDATA)>  
<!ELEMENT codePostal (#PCDATA)>  
<!ELEMENT ville (#PCDATA)>  
<!ELEMENT url (#PCDATA)>  
<!ELEMENT tel (#PCDATA)>  
<!ATTLIST tel t (fixe|fax|mob) "fixe">
```

? : élément optionnel

* : élément optionnel et répétable à l'infini

+ : élément obligatoire et répétable à l'infini

Utilisation d'une DTD locale

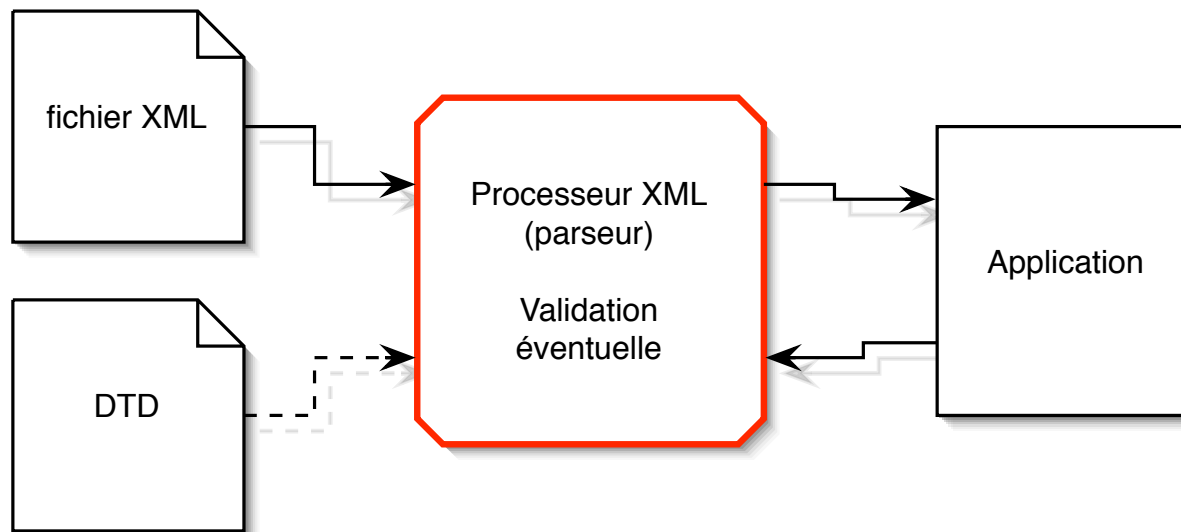
```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE coordonnees SYSTEM "adresse.dtd">
<coordonnees>                                     déclaration de type de document
  <adresse>
    <lignesAdresse>
      <ligne>Centre de Recherche en Informatique</ligne>
      <ligne>Ecole des mines de Paris</ligne>
      <ligne>35, rue Saint-Honoré</ligne>
    </lignesAdresse>
    <codePostal>77305</codePostal>
    <ville>FONTAINEBLEAU Cedex</ville>
  </adresse>
  <url>http://www.cri.ensmp.fr</url>
  <tel t="fixe">01 64 69 47 08</tel>
  <tel t="fax">01 64 69 48 47</tel>
</coordonnees>
```

Utilisation d'une DTD

```
<?xml version="1.0" encoding="iso-8859-1"?> Formal Public Identifier (FPI)  
<!DOCTYPE coordonnees PUBLIC "-//silber//DTD adresse//FR"  
  "http://www.cri.ensmp.fr/people/silber/xml/adresse.dtd">  
<coordonnees> URL de secours  
  <adresse>  
    <lignesAdresse>  
      <ligne>Centre de Recherche en Informatique</ligne>  
      <ligne>Ecole des mines de Paris</ligne>  
      <ligne>35, rue Saint-Honoré</ligne>  
    </lignesAdresse>  
    <codePostal>77305</codePostal>  
    <ville>FONTAINEBLEAU Cedex</ville>  
  </adresse>  
  <url>http://www.cri.ensmp.fr</url>  
  <tel t="fixe">01 64 69 47 08</tel>  
  <tel t="fax">01 64 69 48 47</tel>  
</coordonnees>
```

Processeur XML

- Un module logiciel appelé processeur XML est utilisé pour lire les documents XML et fournir un accès à son contenu (avec validation éventuelle).
- Un processeur XML effectue le travail pour un autre module, l'application.



Exemple d'outil: xmllint

- 'xmllint': un simple parseur validant (xmlsoft.org)

```
champagny:~$ xmllint --valid adresse-avec-dtd.xml
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE coordonnees SYSTEM "adresse.dtd">
<coordonnees>
  <adresse>
    <lignesAdresse>
      <ligne>Centre de Recherche en Informatique</ligne>
      <ligne>Ecole des mines de Paris</ligne>
      <ligne>35, rue Saint-Honoré</ligne>
    </lignesAdresse>
    <codePostal>77305</codePostal>
    <ville>FONTAINEBLEAU Cedex</ville>
  </adresse>
  <url>http://www.cri.ensmp.fr</url>
  <tel t="fixe">01 64 69 47 08</tel>
  <tel t="fax">01 64 69 48 47</tel>
</coordonnees>
champagny:~$
```

Objectifs de conception (I)

- Utilisation facile sur Internet
- Permettre de nombreuses applications
- Compatible avec SGML

Objectifs de conception (2)

- Nombre d'options dans XML réduit au minimum, idéalement aucune
- Documents XML lisibles par l'homme et raisonnablement clairs
- Description de XML formelle et concise

Objectifs de conception (3)

- Facilité d'écriture des programmes traitant les documents XML
- Facilité de création de documents XML
- Concision dans le balisage importe peu

Normes associées

- Unicode et ISO/CEI 10646 pour les caractères (95000 caractères possibles)
- La RFC Internet 1766 pour les noms de langue
- La norme ISO 3166 pour les noms de pays

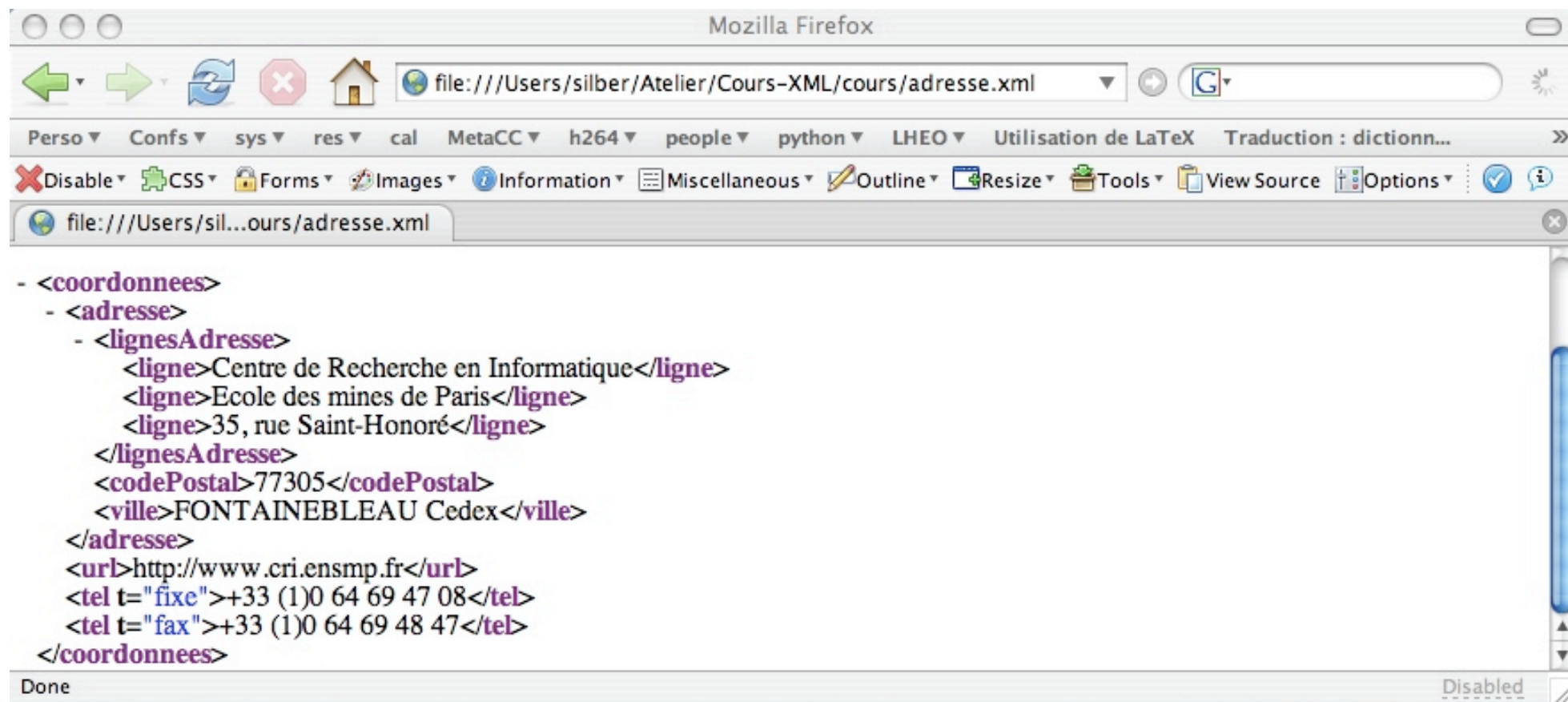
Format ouvert

- Pas de droits à payer pour l'utilisation de XML
- La spécification de XML peut être distribuée librement, à condition que tout le texte et les notices juridiques demeurent intacts

Outils d'édition

Voir du XML

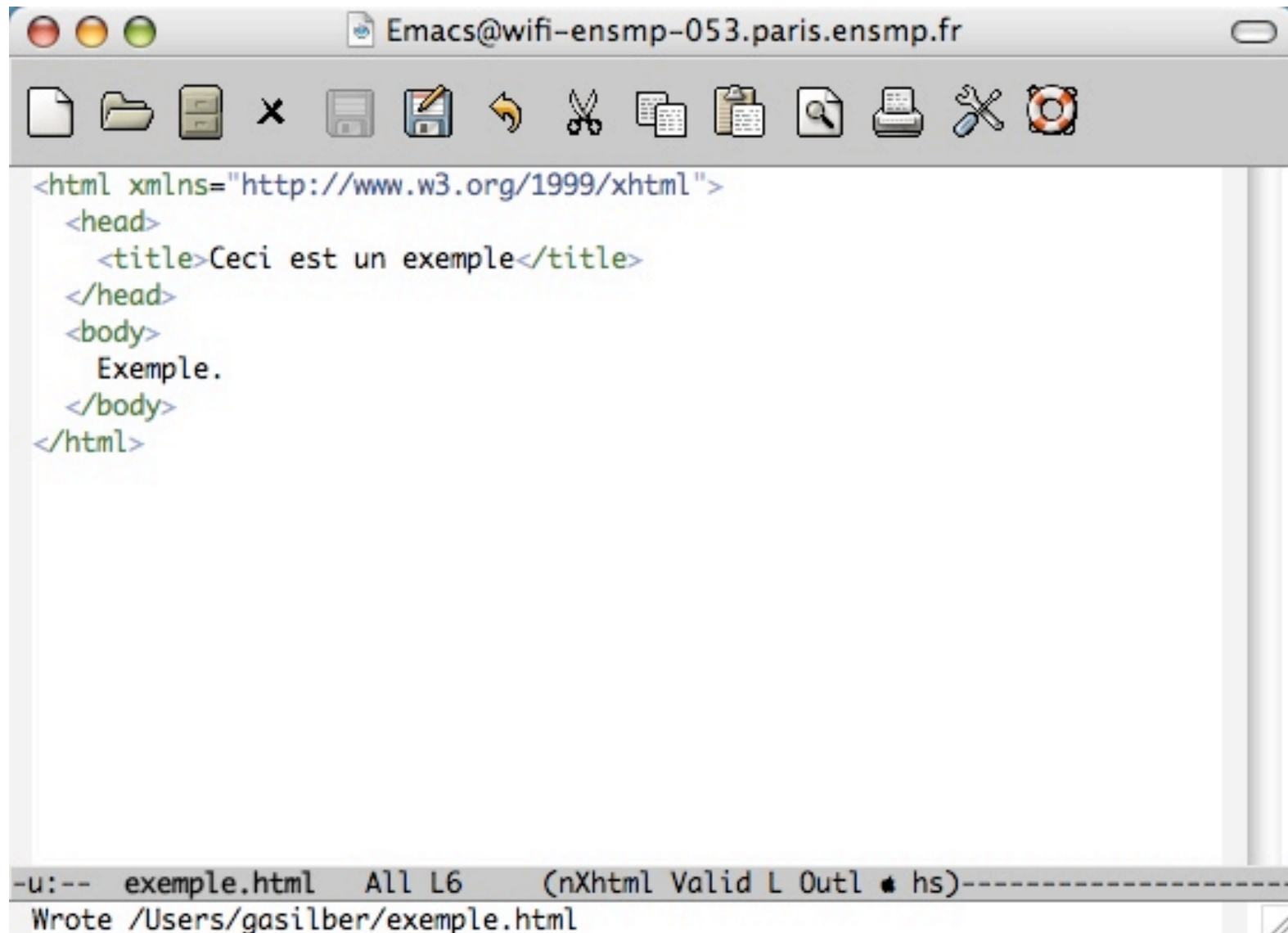
- Editeur de texte (Emacs, notepad, ...)
- Navigateur Web (Firefox, Explorer, ...)



Edition de XML

- Un document XML est du texte, donc n'importe quel éditeur de texte peut être utilisé pour créer et éditer un document XML
 - GNU emacs (www.thaiopensource.com/nxml-mode)
- Il existe néanmoins des éditeurs XML graphiques pour aider à l'édition
 - Xerlin (www.xerlin.org), open source, écrit en Java

Emacs + nxml-mode



The screenshot shows the Emacs editor window titled "Emacs@wifi-ensmp-053.paris.ensmp.fr". The window contains a toolbar with various icons for file operations and editing. The main editing area displays the following XML code:

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Ceci est un exemple</title>
  </head>
  <body>
    Exemple.
  </body>
</html>
```

The status bar at the bottom of the window shows the current file name "exemple.html", the cursor position "All L6", and the mode "(nXhtml Valid L Outl * hs)". Below the status bar, a message indicates "Wrote /Users/gasilber/exemple.html".

Xerlin

The screenshot displays the Xerlin XML editor interface. The main window, titled "Xerlin", contains a toolbar with standard editing icons. Below it, a secondary window titled "adresse-avec-dtd.xml" shows the XML document structure. On the left, a tree view lists the elements and their names. The "tel" element is selected, and its details are shown on the right.

Elément	Nom
▼ coordonnees	
▼ adresse	
▼ lignesAdresse	
▼ ligne	#text
Centre de Recherche en Informatique	
▼ ligne	#text
Ecole des mines de Paris	
▼ ligne	#text
35, rue Saint-Honoré	
▼ codePostal	#text
77305	
▼ ville	#text
FONTAINEBLEAU Cedex	
▼ url	#text
http://www.cri.ensmp.fr	
▼ tel	#text
+33 (1)0 64 69 47 08	
▼ tel	#text
+33 (1)0 64 69 48 47	

On the right side of the editor, the selected "tel" element is shown with a type dropdown set to "fixe" and a text area containing the value "+33 (1)0 64 69 47 08".

eXtensible Stylesheet Language (XSL)

Trois parties distinctes

- XSL Transformations (XSLT)
 - un langage pour transformer du XML
- XML Path Language (XPath)
 - un langage d'expression utilisé par XSLT pour accéder ou se référer à des morceaux de documents XML
- XSL Formatting Objects (XSL-FO)
 - une grammaire XML pour spécifier le formatage des documents

Langage XSLT

- XSLT (XML Stylesheet Language Transformations)
- Langage permettant de transformer des documents XML en d'autres documents XML
- Peut-être vu comme un langage de feuilles de style évolué
- Un document XSLT est lui-même en XML



XSL Transformations (XSLT) Version 1.0

W3C Recommendation 16 November 1999

This version:

<http://www.w3.org/TR/1999/REC-xslt-19991116>
(available in [XML](#) or [HTML](#))

Latest version:

<http://www.w3.org/TR/xslt>

Previous versions:

<http://www.w3.org/TR/1999/PR-xslt-19991008>
<http://www.w3.org/1999/08/WD-xslt-19990813>
<http://www.w3.org/1999/07/WD-xslt-19990709>
<http://www.w3.org/TR/1999/WD-xslt-19990421>
<http://www.w3.org/TR/1998/WD-xsl-19981216>
<http://www.w3.org/TR/1998/WD-xsl-19980818>

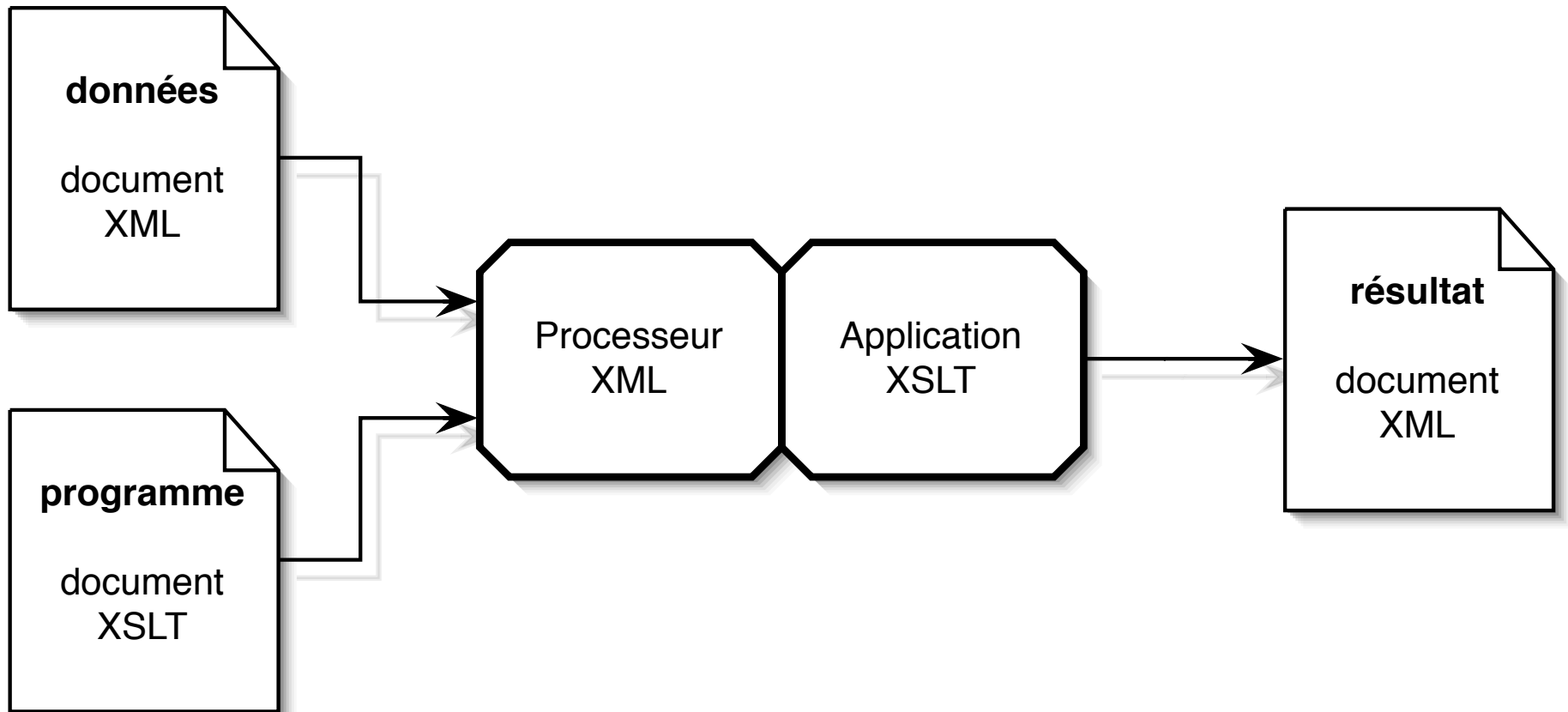
Editor:

James Clark <jjc@jclark.com>

Principe de XSLT

- Lecture du document XML source en mémoire
- Lecture du document XSLT en mémoire
- Application des règles XSLT sur le document source
- Obtention d'un résultat

Chaîne de traitement



Applications XSLT

- Il existe de nombreuses implémentations de XSLT, sous la forme de bibliothèques ou d'outils
- Bibliothèques/outils
 - libxslt2 (xmlsoft.org/XSLT), outil 'xsltproc'
 - xalan (xml.apache.org/xalan-j)
 - saxon (saxon.sourceforge.net)
 - Navigateurs Web modernes

Utilisation de XSLT

- Statiquement, sur le serveur, pour les traitements lourds de documents XML (xsltproc, saxon)
- Dynamiquement, sur le serveur pour le XML généré dynamiquement (xalan)
- Dynamiquement, sur le navigateur client, avec une instruction de mise en page rajoutée dans le XML

```
<?xml-stylesheet href="style.xsl" type="application/xml"?>
```


Exemple: XML vers texte

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">

  <xsl:output method="text"/>

  <xsl:template match="/">
    <xsl:apply-templates select="*" />
  </xsl:template>

  <xsl:template match="tel">
    <xsl:value-of select="@t" />
    <xsl:text>: </xsl:text>
    <xsl:value-of select="text()" />
  </xsl:template>

</xsl:stylesheet>
```

Exemple: XML vers HTML

```
<?xml version="1.0" encoding="iso-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
  <xsl:output method="html" encoding="iso-8859-1"/>
  <xsl:template match="/">
    <html><body><xsl:apply-templates /></body></html>
  </xsl:template>
  <xsl:template match="ligne | ville">
    <code><xsl:value-of select="text()"/></code><br/>
  </xsl:template>
  <xsl:template match="codePostal">
    <code><xsl:value-of select="text()"/></code>
  </xsl:template>
  <xsl:template match="tel">
    <b><xsl:value-of select="@t"/>: </b>
    <code><xsl:value-of select="text()"/></code><br/>
  </xsl:template>
  <xsl:template match="url">
    <a href="{text()}"><xsl:value-of select="text()"/></a><br/>
  </xsl:template>
</xsl:stylesheet>
```

XSL-FO

- Hypothèse: on veut obtenir plusieurs types de documents affichables depuis un document XML (HTML, WML, PDF, texte, RTF, ...)
- Plutôt que d'écrire une transformation XSLT par type de document affichable que l'on veut obtenir, il suffit d'écrire une transformation XSLT qui génère du XSL-FO
- Il existe ensuite des transformations XSLT toutes faites pour convertir le XSL-FO vers d'autres formats

Interfaces de programmation XML

DOM

- Document Object Model
- Héritage de HTML
- Permet de construire un arbre en mémoire depuis un document XML
- Méthodes d'accès et de modification de cet arbre
- <http://www.w3.org/DOM>

SAX

- Simple API for XML
- Lecture séquentielle du document, pas de création d'arbre en mémoire
- Association d'actions (fonctions utilisateur) à la lecture des éléments XML, déclenchées au fur et à mesure de la lecture du document
- <http://www.saxproject.org>

Implémentation

- Il existe une implémentation de SAX et de DOM dans tous les langages courants
- En standard dans JavaEE et dans .NET

Schémas XML

XML Schema

- Norme du W3C
- Permet de représenter une grammaire en XML
- Typage des éléments
- Contrôle du contenu des éléments et des attributs
- Voir exemple sur site web support

Relax NG

- Norme indépendante du W3C
- Permet de représenter une grammaire en XML
- Contrôle du contenu des éléments et des attributs
- Plus simple que les schémas XML
- Voir exemple sur site web support

XML et les bases de données

Stockage du XML

- Système de fichiers
- SGBD Relationnel
- XML Natif

Systeme de fichiers

- Fichiers stockés directement sur le système de fichiers
- Avantages: implémentation aisée, utile pour un petit volume de données
- Inconvénients: l'accès et la mise à jour sont difficiles

SGBDR

- Format de stockage: tables
- Fichiers stockés dans un SGBD sous la forme de Blobs (Binary large objects) ou de Clogs (Character large objects)
- Support de XPath dans certains SGBD comme MySQL
- Avantages: passage à l'échelle, fiabilité et implémentation facile
- Inconvénients: requiert beaucoup de jointures à cause de l'éclatement des documents, nécessité de créer des clés artificielles

XML Natif

- Base de données supportant de XML natif
- Utilisation de modèles de données XML (Schémas)
- Avantages: flexibilité, amélioration des performances d'accès
- Inconvénients: moins mature que les SGBD classiques

Produits XML Natif

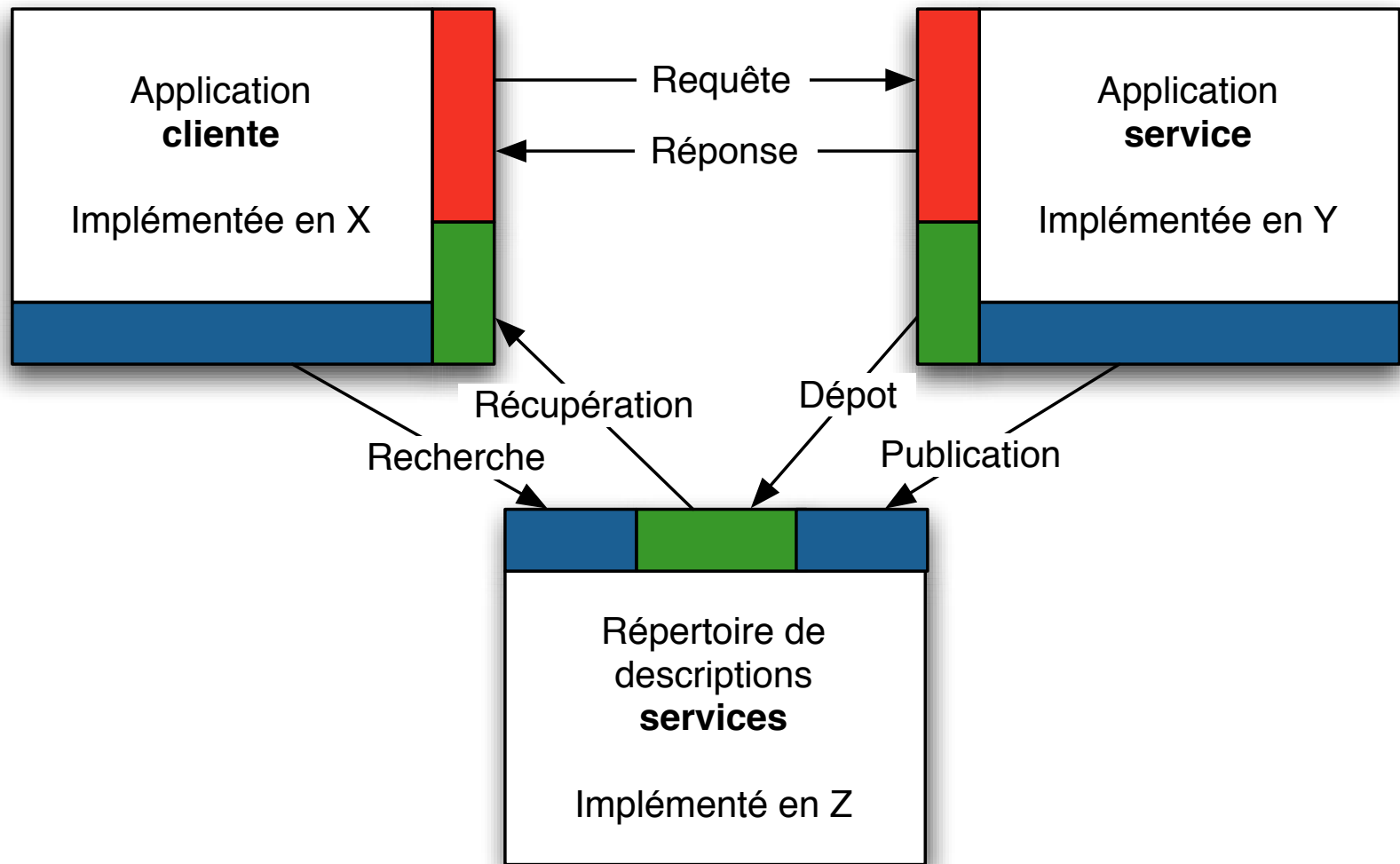
- www.ipedo.com
- xml.apache.org/xindice
- exist.sourceforge.net
- www.softwareag.com/corporate/products/tamino
- www.dbxml.com
- www.cs.toronto.edu/tox
- www.sleepycat.com/products/xml.html (BerkeleyDB XML)

Architectures Orientées Services


Architectures Orientées Service

- Un style d'architecture qui encourage la création de **services** faiblement couplés.
- Ces services peuvent travailler ensemble, sont indépendants de la technologie des **clients**.
- Chaque service fournit une **description** de ce qu'il fait et de comment il peut être utilisé.
- Ces descriptions peuvent être centralisées et découvertes **dynamiquement**.

Principes techniques



 XML, XML-RPC, SOAP, HTTP

 XML, WSDL, HTTP

 XML, Mimine 1.0, (UDDI)

Web Services

Définition en 5 points

- Service disponible sur Internet
(ou sur un {ex,in}tranet)
- N'est lié à aucun système d'exploitation/matériel
- Utilise un système de messages standardisé
« à la XML »
- S'auto-décrit grâce à une grammaire XML
- Peut-être découvert grâce à un mécanisme de recherche

Technologies

- Technologies internet (TCP/IP, HTTP, SMTP, FTP) pour le service disponible sur Internet
- Documents envoyés : XML-RPC, SOAP, XML, JSON
- Description du service : WSDL
- Recherche du service : UDDI

Messagerie

- XML-RPC (www.xmlrpc.com)
- Protocole simple, types de données simples, pas un standard W3C
- SOAP (Simple Object Access Protocol)
- Norme du W3C, séparation enveloppe/données, types de données complexes

Description

- WSDL (Web Service Description Language)
- Langage XML de description de Web Service
- Standard du W3C utilisant SOAP et les Schémas XML

UDDI

- UDDI (www.uddi.org)
- Universal Description, Discovery and Integration
- Service permettant de rechercher des services et de récupérer automatiquement leur description (par exemple en WSDL)

Examples

- Google API
 - <http://api.google.com>
- NEOS Server for optimization
 - <http://www-neos.mcs.anl.gov/neos>
- Amazon Web Services
 - <http://www.amazon.com/gp/aws/landing.html>

REST

- REpresentational State Transfer
- Présenté en 2000 par Roy Fielding (HTTP, Apache)
- Pas un standard mais un style d'architecture web
- Le Web est un système REST !
- Mode d'utilisation des standards existants
 - HTTP, URL, XML, HTML, MIME

Atom et RSS

- Deux formats de syndication
 - RSS 2.0 (Really Simple Syndication)
 - Atom 1.0
- Un protocole “RESTful”:APP
 - Atom Publishing Protocol (HTTP)