

Cocoon

<http://cocoon.apache.org/>

(alain.bouju@univ-lr.fr)
(<http://perso.univ-lr.fr/abouju>)

Présentation

Apache Cocoon est un système de développement web permettant une séparation des problèmes et un développement sous la forme de composants.

Le système utilise la notion de pipeline.

Dans ce pipeline chaque composant effectue une opération.

Historique – Cocoon 1

- Cocoon est un projet que Stephano Mazzocchi a débuté en Janvier 1999,
- le projet a commencé par un servlet permettant la transformation de contenu XML par des transformations XSL,
- le projet a utilisé l'API DOM.

Bref historique

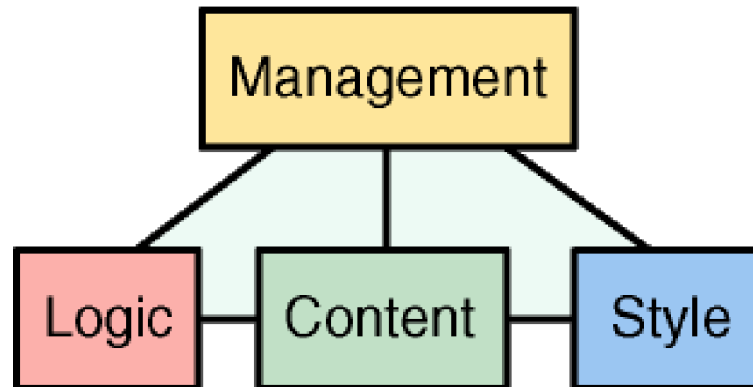
Amélioration des performances par l'utilisation d'un parser SAX,

Une organisation autour d'un fichier XML sitemap décrivant le comportement du site et l'organisation du pipeline,

De meilleure performance, par pré-compilation, pré-génération et cache.

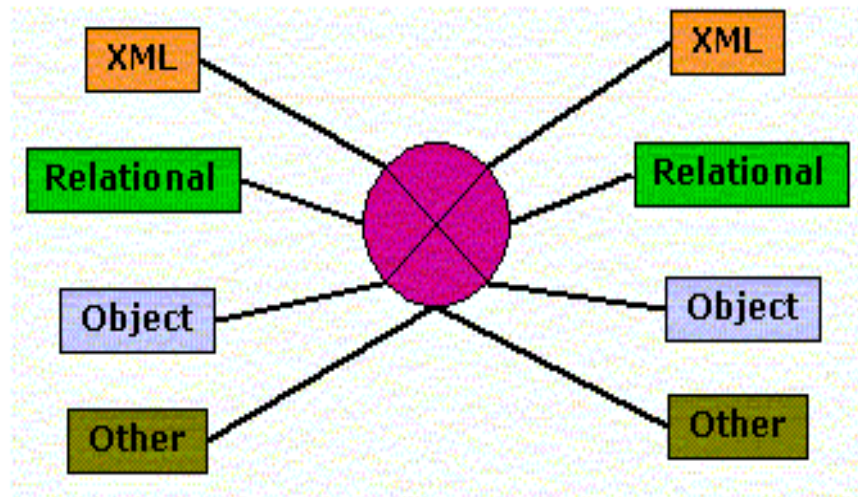
Objectifs

Séparation du contenu, du style et de la logique



Manipulation de données

Le système permet de manipuler et présenter un ensemble de type de données.

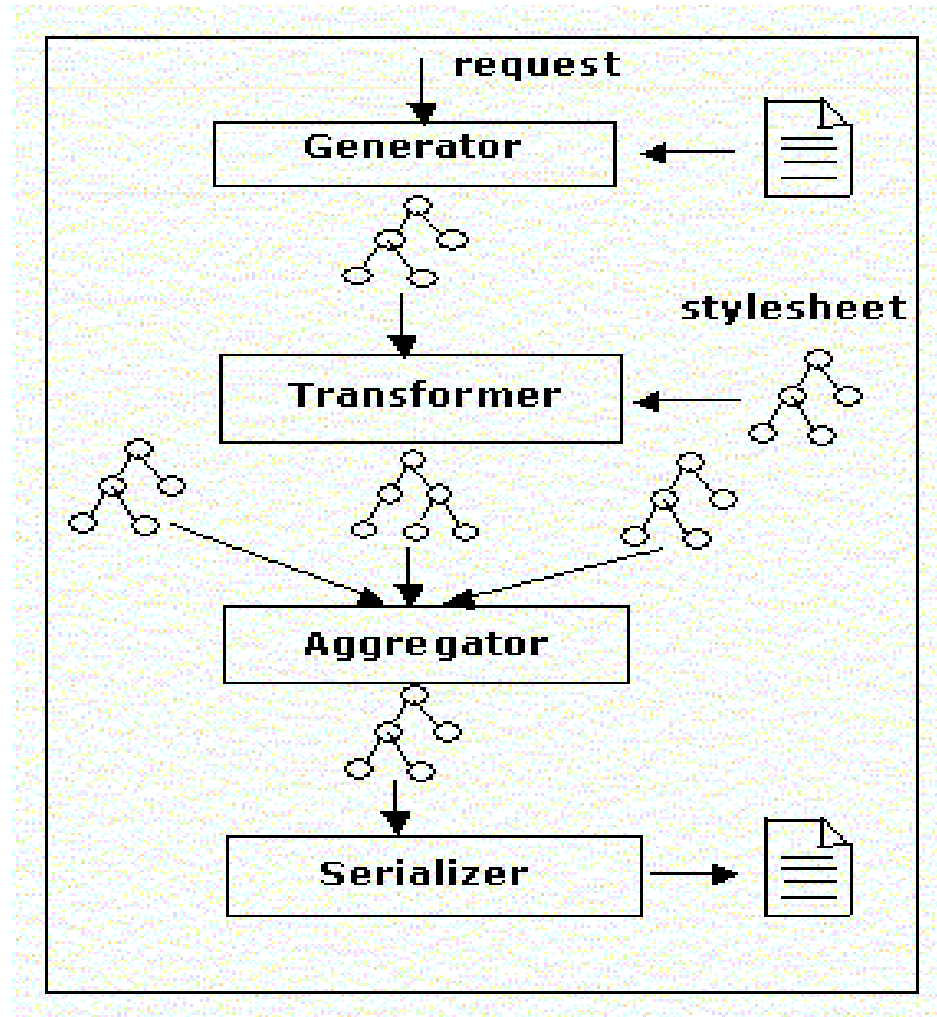


Mécanismes

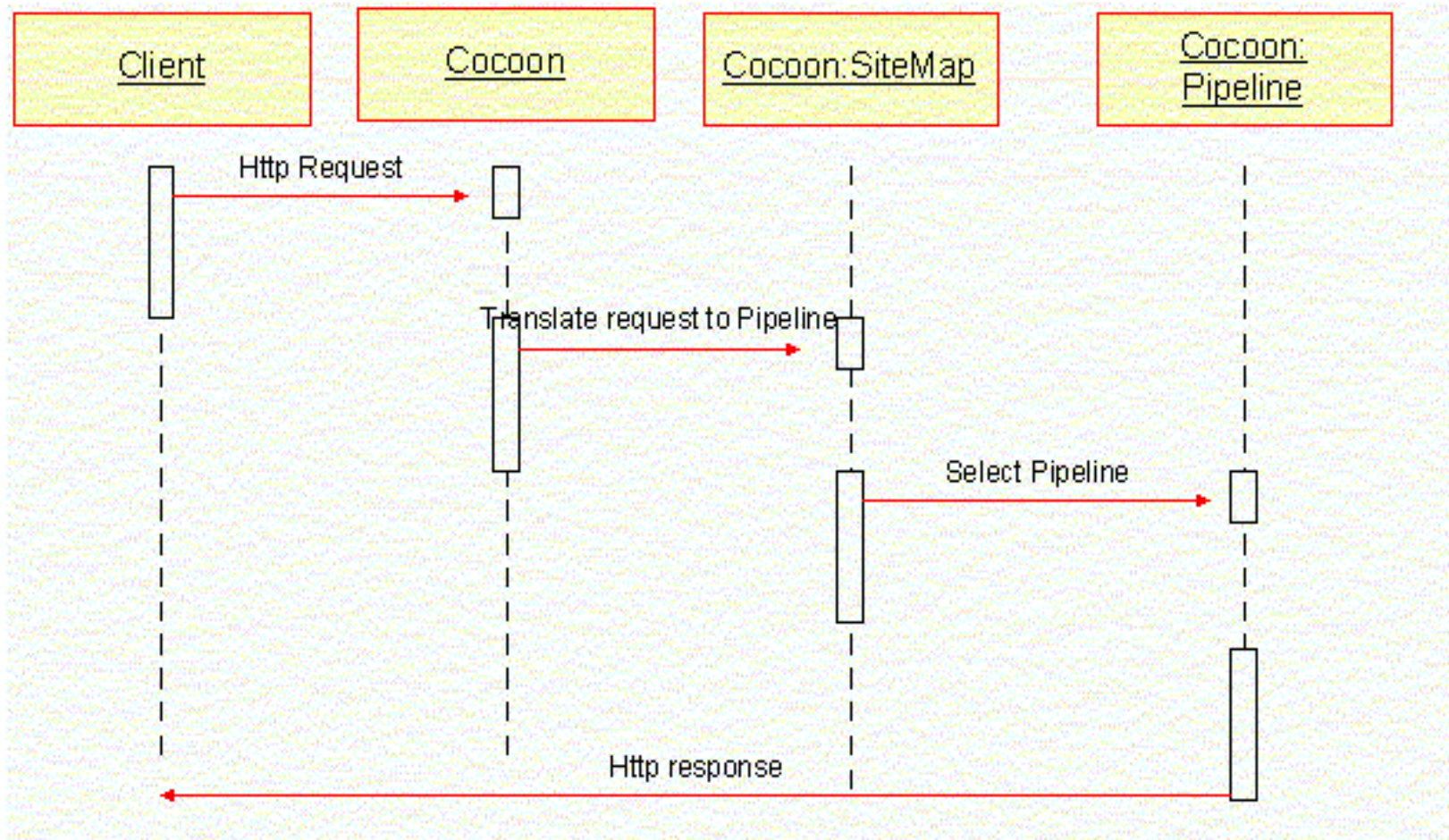
Mécanisme de bases pour des documents XMLs

- répartition par des "Matchers",
- génération de documents XML par des "Generators",
- transformation de documents XML par des "Transformers",
- aggrégation par des "Aggregators",
- production du document final par des "Serializers".

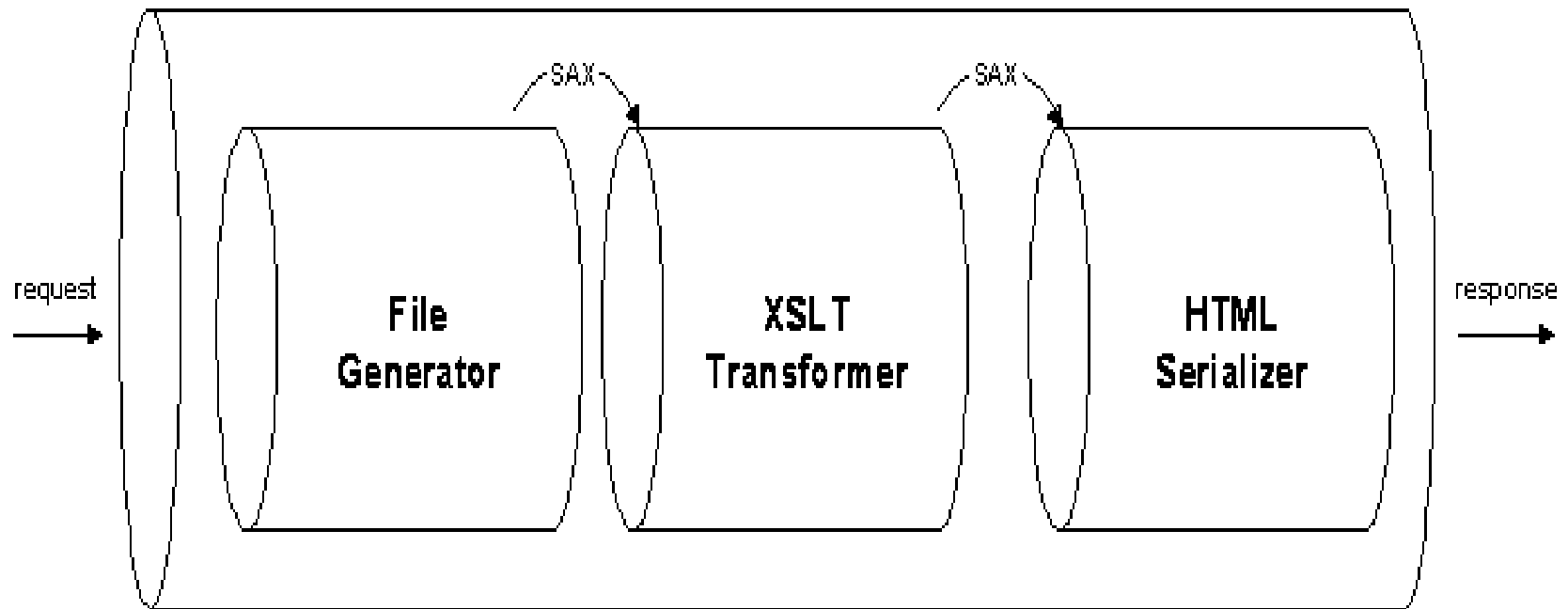
Mécanismes



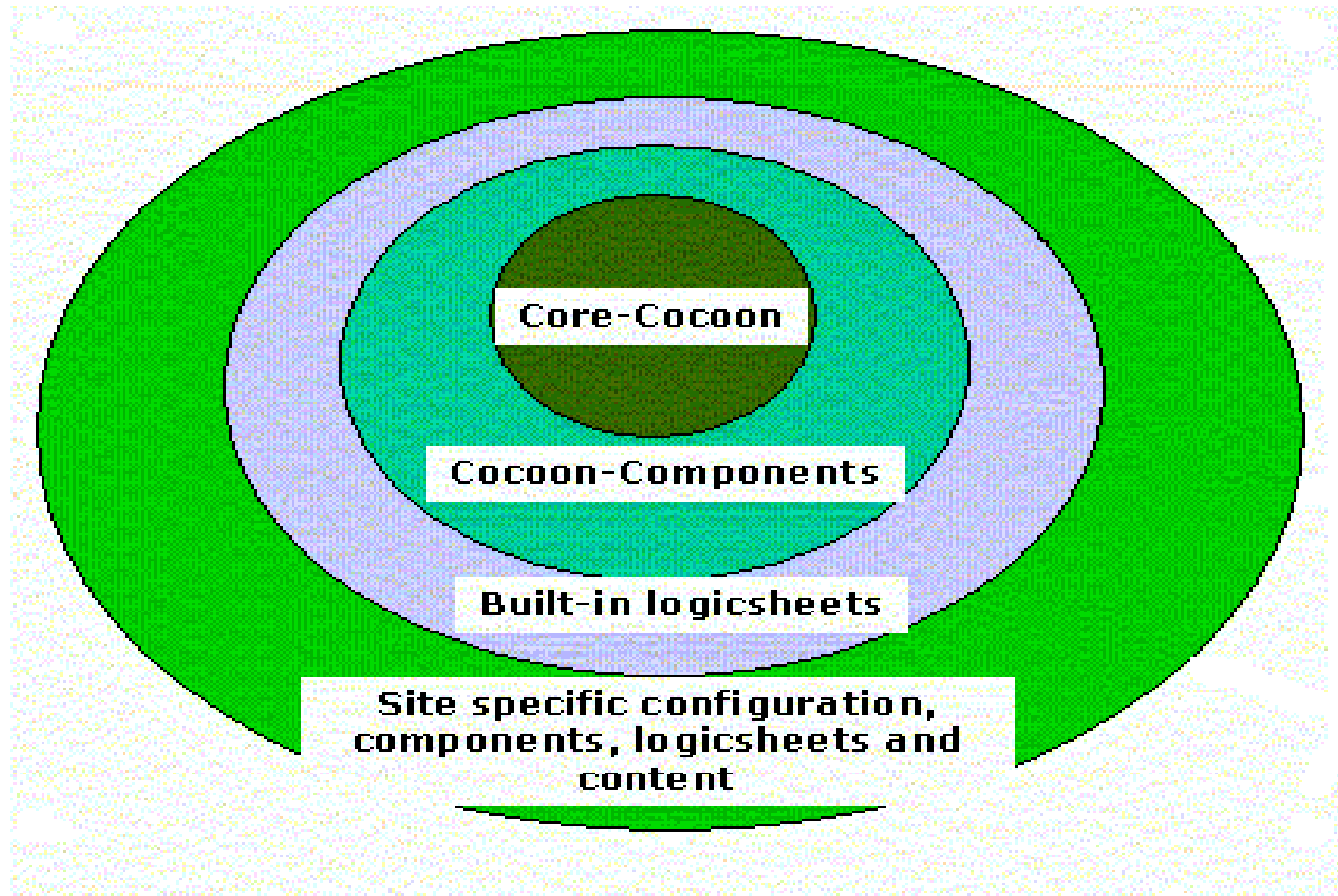
Gestion requête



Pipeline



Architecture



Core-cocoon

- mécanisme de thread,
- gestion contexte,
- mécanisme de trace (log),
- configuration

Components

- des composants générateurs (generator),
- des composants de transformation,
- des composants d'association (matchers),
- des composants de sérialisation
- ...

Des fichiers de configuration

- sitemap.xmap,
- xsp.xsl
- request.xsl,
- response.xsl,
- ...

Components

- des composants générateurs (generator),
- des composants de transformation,
- des composants d'association (matchers),
- des composants de sérialisation
- ...

eXtensible Server Pages (XSPs)

Une page XSP est une page XML avec les éléments suivants :

- La racine du document doit être `<xsp:page>`,
- Il y a la déclaration du langage comme attribut de `<xsp:page>`,
- Il doit y avoir une déclaration de "namespace" comme attribut de `<xsp:page>`,
- Pour avoir un XSP utile il faut au moins les éléments `<xsp:logic>` et `<xsp:expr>`.

Cocoon - eXtensible Server Pages (XSPs)

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsp:page language="java"
  xmlns:xsp="http://apache.org/xsp">
  <xsp:logic>
    static private int counter = 0;
    private synchronized int count()
    { return counter++;
    }
  </xsp:logic>
  <page>
    <p>I have been requested
    <xsp:expr>count()</xsp:expr> times.</p>
  </page>
</xsp:page>
```

Cocoon – Code Java XSP

```
package org.apache.cocoon.www.docs.samples.xsp;

import java.io.File;

// autres imports

public class counter_xsp extends XSPGenerator {

    // .. Bookkeeping stuff commented out.

    /* User Class Declarations */

    static private int counter = 0;

    private synchronized int count() {

        return counter++;

    }

    /* Generate XML data. */

    public void generate() throws SAXException {

        this.contentHandler.startDocument();

        AttributesImpl xspAttr = new AttributesImpl();

        this.contentHandler.startPrefixMapping("xsp", "http://apache.org/xsp");

        this.contentHandler.startElement("", "page", "page", xspAttr);

        // Statements to build the XML document (Omitted)

        this.contentHandler.endElement("", "page", "page");

        this.contentHandler.endPrefixMapping("xsp");

        this.contentHandler.endDocument();

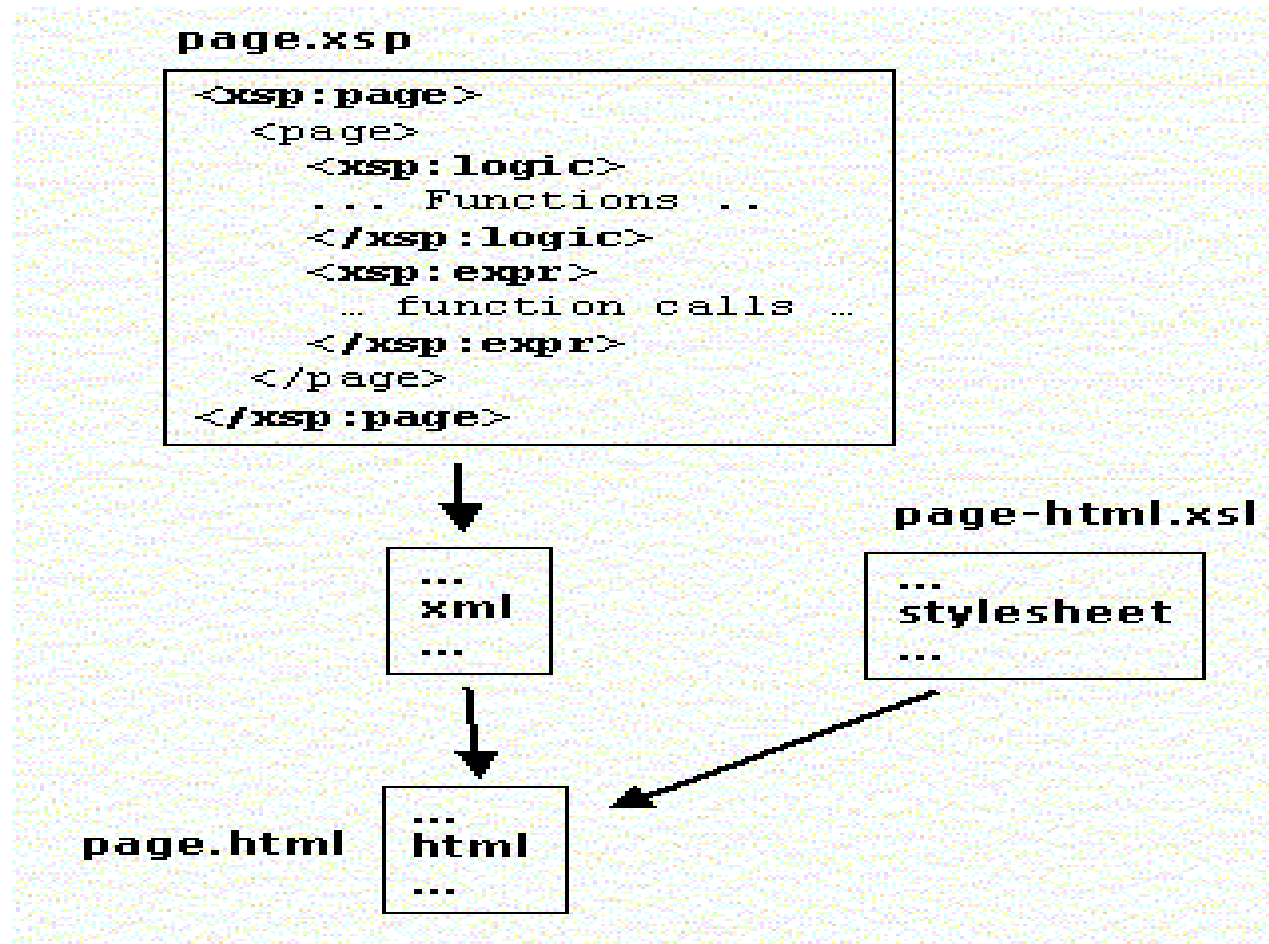
    }

}
```

Ecriture XSP Cas Simple

- le code est dans la page xml,
- pas de séparation entre le contenu et la logique,
- pour de petits exemples

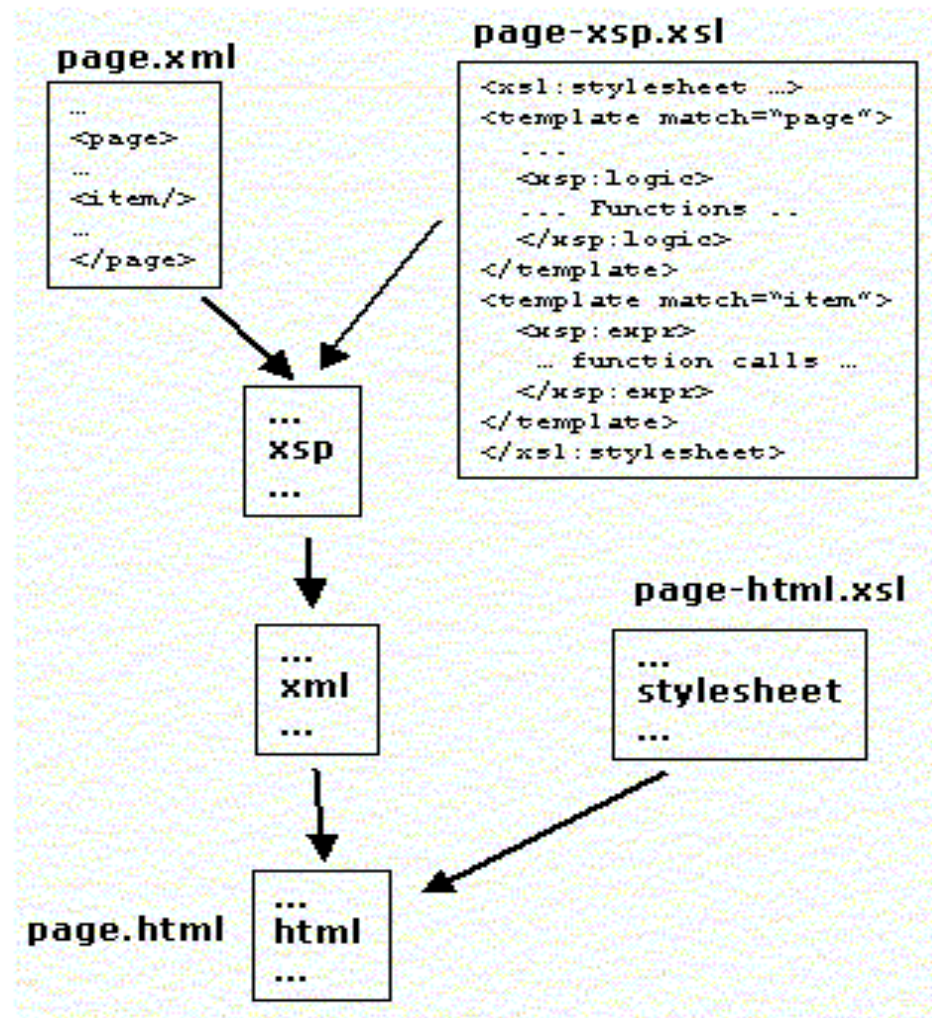
Architecture XSP



Utilisation XSLT

- le code est dans une feuille de style xsl,
- on a une séparation entre le contenu et la logique,
- meilleure technique de création d'XSP.

Utilisation XSLT



Cocoon – Sitemap

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<map:sitemap xmlns:map="http://apache.org/cocoon/sitemap/1.0">
  <map:components>
    ...
  </map:components>
  <map:views>...</map:views>
  <map:pipelines>
    <map:pipeline>
      <map:match> ...</map:match> ...
    </map:pipeline>
    ...
  </map:pipelines>
  ...
</map:sitemap>
```

Le fichier sitemap contient des informations de configuration pour cocoon

- une liste de composants d'association (matchers),
- une liste de composants générateurs (generator)
- une liste de composants de transformation (transformers),
- une liste de composants de lecture (readers)
- une liste de composants de sérialisation (serializers)
- une liste de composants de choix (selectors)
- une liste de composants pipelines associé à des conditions (patterns)
- ...

Matcher

Composant d'association

Un composant essaye d'associer un URI avec des conditions (pattern) pour associer la requête à un "pipeline" de traitement.

Il y a différents types de composants :

- composant d'association "wildcard",
- composant d'association "regexp",

Des composants peuvent être ajoutés sans modifier Cocoon.

Le composant permet d'associer un traitement spécifique pour un groupe d'URI.

Matcher

Composant d'association

Insertion de composants d'association (matcher) dans un sitemap.

```
<map:matchers default="wildcard">
  <map:matcher name="wildcard"
    factory="org.apache.cocoon.matching.WildcardURIMatcher"/>
  <map:matcher name="regexp"
    factory="org.apache.cocoon.matching.RegexpURIMatcher"/>
</map:matchers>
```

Utilisation de composants d'association pour des pipelines

```
<map:match pattern="jsp/*">
  <map:generate type="jsp" src="/docs/samples/jsp/{1}.jsp"/>
  ...
</map:match>
<map:match pattern="hello.pdf">
</map:match>
```

Generator - Composant générateur

Un composant générateur est utilisé pour créer une structure depuis une source d'entrée (fichier, répertoire, flux, ...).

Il y a différents types de composants

- composant "fichier",
- composant "répertoire",
- composant "XSP",
- composant "JSP",

Des composants peuvent être ajoutés sans modifier Cocoon.

Generator - Composant générateur

Insertion de composants générateurs (generator) dans un sitemap.

```
<map:generators default="file">
  <map:generator name="file"
    src="org.apache.cocoon.generation.FileGenerator"
    label="content"/>
  <map:generator name="directory"
    src="org.apache.cocoon.generation.DirectoryGenerator"
    label="content"/>
  <map:generator name="serverpages"
    src="org.apache.cocoon.generation.ServerPagesGenerator"
    label="content"/>
  ...
</map:generators>
```

Transformer - Composant de transformation

Un composant de transformation permet de passer d'un flux XML vers un autre,

Il y a différents types de composants

- composant "XSLT",
- composant "Log",
- composant "SQL",
- composant "I18N",
-

Des composants peuvent être ajoutés sans modifier Cocoon.

Transformer - Composant de transformation

Insertion de composant de transformation (Transformer) dans un sitemap.

```
<map:match pattern="hello.html">  
  <map:generate src="docs/samples/hello-page.xml"/>  
  <map:transform src="stylesheets/page/simple-page2html.xsl"/>  
  <map:serialize type="html"/>  
</map:match>
```

Un composant simple de transformation XSL.

Serializer - Composant de « sérialisation »

Un composant de sérialisation est utilisé pour transformer un format d'entré XML en un autre format (pas nécessairement XML).

Il y a différents types de composants

- composant "HTML",
- composant "FOP",
- composant "Text",
- composant "XML",
- ...

Des composants peuvent être ajoutés sans modifier Cocoon.

Serializer - Composant de « sérialisation »

Insertion de composants de sérialisation (Serializer) dans un sitemap.

```
<map:serializers default="html">
  <map:serializer name="xml" mime-type="text/xml"
    src="org.apache.cocoon.serialization.XMLSerializer"/>
  <map:serializer name="html" mime-type="text/html"
    src="org.apache.cocoon.serialization.HTMLSerializer"/>
  <map:serializer name="fo2pdf" mime-type="application/pdf"
    src="org.apache.cocoon.serialization.FOPSerializer"/>
  <map:serializer name="vrml" mime-type="model/vrml"
    src="org.apache.cocoon.serialization.TextSerializer"/>
  ...
</map:serializers>
```

Serializer - Composant de « sérialisation »

Insertion d'un composant de sérialisation (Serializer) dans un pipeline.

```
<map:match pattern="hello.html">  
  <map:generate src="docs/samples/hello-page.xml"/>  
  <map:transform src="stylesheets/page/simple-page2html.xsl"/>  
  <map:serialize type="html"/>  
</map:match>
```

Pipeline

Le fichier de configuration sitemap permet de mettre en place des "pipelines" composés de composants de génération, de transformation, de sérialisation, ...

Les requêtes sont réparties en fonction d'association (URI, ...) avec les pipelines.

Le pipeline est réalisé à partir du fichier sitemap.xmap

Pipeline

