

# INF157 - Utilisation des Réseaux

## Licence 3 Informatique

Arnaud Pecher (repris par Damien Magoni)

Bureau 322, Bâtiment A30, LaBRI  
Université de Bordeaux

Licence 3 Informatique - Bordeaux

- 1 Les WIKIs ... selon Wikipedia !
- 2 Un WIKI : pmwiki
- 3 Le projet
  - Le code de base

## *Bibliographie*

- Wikipedia ;-)
- wikimatrix [http ://www.wikimatrix.org/](http://www.wikimatrix.org/)

1 Les WIKIs ... selon Wikipedia !

2 Un WIKI : pmwiki

3 Le projet

- Le code de base

## Wiki

Un wiki est un système de gestion de contenu de site Web qui rend les pages Web librement modifiables par tous les visiteurs autorisés.

- Faciliter l'écriture collaborative de documents avec un minimum de contraintes.
- Inventé en 1995 par Ward Cunningham.
- Le mot « wiki » vient du redoublement hawaïen wiki wiki, qui signifie « rapide ».

## Moteur de wiki

Logiciel qui met en œuvre la gestion du contenu des pages.

- Un wiki n'est pas forcément modifiable par tout le monde ; on peut exiger que les visiteurs s'inscrivent avant d'être autorisés à modifier les pages.
- Lorsqu'un wiki autorise des visiteurs anonymes à modifier les pages, c'est l'adresse IP de ces derniers qui les identifie

- On accède à un wiki, en lecture comme en écriture, avec un navigateur Web classique.
- Deux modes différents : le mode lecture, qui est le mode par défaut, et le mode édition, qui présente la page sous une forme qui permet de la modifier.
- En mode édition, le texte de la page, affiché dans un formulaire Web, s'enrichit d'un certain nombre de caractères supplémentaires, suivant les règles d'une syntaxe particulière : le wikitexte, qui permet d'indiquer la mise en forme du texte, de créer des liens, de disposer des images, etc.
- Quelques wikis proposent une interface d'édition WYSIWYG.

## Création des pages :

- Aucune contrainte organisationnelle.
- Les WikiWords (mots avec plusieurs capitales) désignent automatiquement une page.

## Suivi des modifications :

- Un wiki incorpore un système de gestion des versions d'une page.
- Le suivi des versions d'une page n'est pas librement modifiable.

# Comparaison des moteurs de wikis

<http://www.wikimatrix.org>

Common Features	MediaWiki	PmWiki
Preview	Yes	Yes
Minor Changes	Yes	Yes
Change Summary	Yes	Yes
Page History	Yes	Yes
Page Revisions	Unlimited	Unlimited
Revision Diffs	Between all	Between all
Page Index	Yes	Yes
Plugin System		
Special Features	MediaWiki	PmWiki
Unicode Support	Yes	Yes
Right-to-Left Support	Yes	
Interface Languages	100+	20+
Email notification	Optional	Optional
Comments	Discussion Pages	Plugin
Categories	Yes	Yes
Namespaces	Yes	Yes
Page Redirection	Yes	Yes
Conflict Handling	Conflict Resolution	Conflict Resolution
Search	Full Text	Full Text
Links	MediaWiki	PmWiki
CamelCase	No	Optional
Freelinks	Yes	Yes
Backlinks	Yes	Yes
InterWiki	Yes	Yes
SisterWiki	No	No
Image Links	Yes	Yes
Windows Shares	No	Yes



1 Les WIKIs ... selon Wikipedia !

2 Un WIKI : pmwiki

3 Le projet

- Le code de base



Page de garde  
Fr  
Bac à sable?

[PmWikiFr /](#)  
**PmWikiFr**

[Vue](#) [Éditer](#) [Historique](#) [Imprimer](#)

## En Français

Fonctionnalités  
Téléchargement  
Installation  
Bases de l'édition  
Documentation  
FAQ PmWikiFr  
Philosophie de PmWiki  
Notes de versions  
Feuille de route

PmWiki est un bon système de type [wiki](#) pour la création et l'entretien collectif de sites Internet.

Les pages PmWiki ont le même aspect et fonctionnent comme des pages Internet ordinaires, sauf qu'elles possèdent un lien pour "[éditer](#)" ce qui permet de modifier ou d'ajouter facilement des pages à un site, en utilisant les règles d'[édition de base](#). Vous n'avez pas besoin de connaître le langage HTML ou les CSS. L'édition des pages peut être laissée ouverte à tout public ou restreinte à un petit groupe d'auteurs.

## Fonctionnalités principales

**Présentation personnalisée:** L'administrateur d'un site peut rapidement modifier l'aspect et les fonctions d'un site PmWiki en utilisant des [Thèmes](#) différents et des modèles HTML. Si aucun [thème existant](#) parmi la bonne douzaine qui existe ne vous convient, vous pouvez aisément en modifier ou en créer un.

**Contrôle d'accès:** Une protection de PmWiki par mot de passe peut être appliquée à un site entier, à des groupes de pages, ou à des pages individuelles. La protection par mot de passe contrôle qui peut lire les pages, qui peut les éditer et qui peut télécharger des attachements. Le système d'accès est intégré à PmWiki mais il peut aussi fonctionner en conjonction avec des bases de mots de passe existantes, telles que `.htaccess`, un serveur LDAP, ou une base de donnée MySQL.

**Personnalisation et modules:** Un des principes de la [Philosophie de PmWiki](#) est d'inclure uniquement les fonctions essentielles au noyau du système. Les administrateurs peuvent aisément personnaliser PmWiki et ajouter de nouvelles balises. Des centaines de fonctions peuvent être installées en utilisant des (appelées "recettes") qui sont disponibles depuis le "[Cookbook](#)".

Doc  
administration  
Doc  
compléments  
Administration  
site

## Traduction

Etat de la  
Traduction

## Pmwiki.org

Liste de

RealOpt

Search  Go

Text Size: - 0 +


**RealOpt**

- Presentation
- Members
- Publications
- Industrial Collaborations
- Software
- Seminars
- Teaching
- Intranet

**Help**

- Basic Editing
- Documentation Index
- FAQ
- Wiki philosophy

edit SideBar



GeminiSkin  
powered by PmWiki

Project

## Welcome to RealOpt web site

### Reformulations based algorithms for Combinatorial Optimization: an INRIA research group in discrete optimization, mathematical programming, and graph theory.



#### News

- Open position:**
  - Professorship in Mathematical Programming
- Special events:
  - ROADEF 2008, February 25 - 27, 2008, Clermont-Ferrand, France (deadlines: long papers, october 21st, 2007 - short papers, november 9th, 2007).
  - International Symposium on Combinatorial Optimization: CO2008, March 16-19, 2008, Coventry, United-Kingdom (deadline november 19th, 2007)
  - 8th Latin American Theoretical Informatics: LATIN 2008, April 7-11, 2008, Rio de Janeiro, Brazil (deadline September 21, 2007)
  - 7th Cologne-Twente Workshop on Graphs and Combinatorial Optimization: CTW 2007, May 13-15th, Gargnano, Italy (deadline February 15th)
  - 13th Conference on Integer Programming and Combinatorial Optimization: IPCO XIII, May 26-28, Bertinoro, Italy (deadline November 15th, 2007)
  - 21st Meeting of the European Chapter on Combinatorial Optimization: ECCO XXI, May 29-31, 2008, Dubrovnik, Croatia (deadline March 9, 2008).

#### Links

- [Inria Web Site](#)
- [Presentation](#)
- [Members](#)
- [Publications](#)
- [Industrial Collaborations](#)
- [Software](#)
- [Seminars of the work group](#)
- [Teaching](#)

#### Contact

- François Vanderbeck: [fv@math.u-bordeaux1.fr](mailto:fv@math.u-bordeaux1.fr)

[Edit](#) [History](#) [Source](#) [Attach File](#) [Backlinks](#) [List Group](#)

Page last modified by October 30, 2007, at 01:08 PM

Navigation icons: back, forward, search, etc.

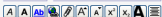


# Vue : édition de la page

## Editing Project>Welcome

Save Save and edit Preview Cancel Author:

Show Help



(:title Welcome to RealOpt web site:)

```
!!!%center% '+%blue%'Re''%formulations based %blue%'a''%gorithms for Combinatorial  
%blue%'Opt''%imization+':
```

```
%center% '+%green%'an INRIA research group in discrete optimization, ''%'+  
%center% '+%green%' mathematical programming, ''%'+  
%center% '+%green%'and graph theory''%'+
```

```
%center% %cfloat height=45px width=120px% [[http://www.inria.fr|Attach:logoInria.png]]
```

!!! News

```
* '+%red%'Open position''%'+:  
**[[http://www.math.u-bordeaux.fr/~fv/pubProfR0/ | Professorship in Mathematical Programming ]]
```

```
* Special events:  
** [[http://www.isima.fr/roadev2008 | ROADEF 2008]], February 25 - 27, 2008, Clermont-Ferrand, France (deadlines:  
long papers, october 21st, 2007 - short papers, november 9th, 2007).  
** International Symposium on Combinatorial Optimization: [[http://go.warwick.ac.uk/C02008/ | C02008]], March  
16-19. 2008. Coventry, United-Kinadom (deadline november 19th. 2007)
```

Summary:

This is a minor edit

Save Save and edit Preview Cancel

Close Preview

End of preview — remember to save — Top — Close Preview

## Character formatting

<p><i>Emphasize, strongly, very strongly.</i></p> <ul style="list-style-type: none"><li>• These are all apostrophes (single-quote marks), not double-quotes.</li></ul>	<pre>"Emphasize", "'strongly'", "''''very strongly''''".</pre>
<p>Monospaced text</p> <p>This is a preformatted text block</p>	<pre>@@Monospaced text@@  [@This is a preformatted text block@]</pre>
<ul style="list-style-type: none"><li>• Text with <sup>superscripts</sup></li><li>• Text with <sub>subscripts</sub></li><li>• deleted <del>strikethrough</del> text</li><li>• inserted <u>underline</u> text</li><li>• big, bigger text</li><li>• small, smaller text</li></ul>	<pre>* Text with '^superscripts^' * Text with '_subscripts_' * deleted (-strikethrough-) text * inserted (+underline+) text * [+big+], [++bigger++] text * [-small-], [--smaller--] text</pre>
<p>WikiStyles allow text to be different colors.</p> <ul style="list-style-type: none"><li>• Use lowercase letters for RGB colors ( #ff7f00, not #FF7F00).</li></ul>	<pre>[[WikiStyles]] allow %red% text %blue% to %green% be %color=#ff7f00% different %%colors.</pre>

## Headings and blocks

<h3>Major Subheading</h3> <hr/> <h3>Major Subheading With Smaller Text</h3> <hr/> <h3>Minor Subheading</h3> <p><i>And More</i></p>	<pre>!! Major Subheading !! [-Major Subheading With Smaller Text-] !!! Minor Subheading !!!! And More !!!!!! Subheadings</pre>
--	--

1 Les WIKIs ... selon Wikipedia !

2 Un WIKI : pmwiki

3 **Le projet**

- Le code de base

Le script `deploy.sh` se charge de compiler la servlet et de la déployer dans tomcat.

## deploy.sh

---

- 1 création de la structure des répertoires de la servlet (via le script `createDirectories.sh`);
- 2 recopie des sources java dans `webapps/src`, compilation, et déplacement des binaires dans `webapps/deploiy/classes` (via le script `compil.sh`);
- 3 recopie du fichier `web.xml` dans `WEB_INF` ;
- 4 recopie du fichier `params.txt` dans `webapps/deploiy` ;
- 5 création d'une archive `.war` pour l'importation dans tomcat ;
- 6 déploiement de la servlet (à partir de l'archive) dans tomcat (et relancement de celui-ci).



# Scripts Unix de déploiement

createDirectories.sh \_\_\_\_\_

```
mkdir webapps && mkdir webapps/src
mkdir webapps/deploy && mkdir webapps/deploy/WEB-INF
mkdir webapps/deploy/WEB-INF/classes
```

compile.sh \_\_\_\_\_

```
javac -cp ./servlet-api.jar webapps/src/*.java
mv webapps/src/*.class webapps/deploy/WEB-INF/classes/
```

deploy.sh \_\_\_\_\_

```
echo "reinitialisation de webapps/deploy (recompilation)"
sh createDirectories.sh && sh compile.sh
echo "recopie du fichier web.xml dans WEB-INF"
cp web.xml webapps/deploy/WEB-INF/
echo "recopie du fichier params.txt dans deploy"
cp params.txt webapps/deploy/
echo "creation de l'archive .war pour tomcat"
cd webapps/deploy && jar cf ../../MaServletWiki.war *
echo "déploiement et relancement de tomcat"
sh $CATALINA_HOME/bin/shutdown.sh
rm -fr $CATALINA_HOME/webapps/MaServletWiki*
cp ../../MaServletWiki.war $CATALINA_HOME/webapps/
sh $CATALINA_HOME/bin/startup.sh
echo "termine."
```

## WikiEngine extends HttpServlet

**Variables** : W\_action, W\_page

**Méthodes** :

- public WikiEngine()
- public void doGet (HttpServletRequest, HttpServletResponse ) throws IOException
- public void doPost (HttpServletRequest, HttpServletResponse res) throws IOException
- private String XHTMLContent()
- private String XHTMLBody()
- private String XHTMLHead( String )
- private String loadFile( String )

### Commentaires

- 1 les 2 variables stockent la page Wiki courrante et le mode de rendu (edit, print etc ...);
- 2 les méthodes doGet et doPost sont appelées par le navigateur, en fonction de la requête HTTP;
- 3 les 3 méthodes XHTMLxxx construisent le code XHTML à retourner au navigateur;
- 4 la méthode loadFile charge le fichier texte correspondant à la page Wiki demandée.

```
public void doGet (HttpServletRequest, HttpServletResponse ) throws IOException
```

```
try { StringBuffer url = req.getRequestURL ();  
W_page = url.substring(url.lastIndexOf("/") + 1);  
// output  
PrintWriter out = res.getWriter ();  
res.setContentType ("text/html; charset=ISO-8859-1");  
W_action = req.getParameter (W_ACTION);  
out.println(XHTMLContent ());  
}  
catch (IOException e){  
res.sendError (HttpServletResponse.SC_SERVICE_UNAVAILABLE); }
```

## Commentaires

- 1 méthode appelée lorsque le client demande l'URL via la méthode HTTP Get (cas général)
- 2 récupération du nom de la page dans W\_page
- 3 récupération du mode d'action dans W\_action
- 4 envoi au client du code XHTML correspondant généré par la méthode XHTMLContent()

```
public void doPost (HttpServletRequest req, HttpServletResponse res)
```

```
try { // sauvegarde du code de la page entree dans le formulaire
StringBuffer urlB = req.getRequestURL ();
String page = urlB.substring (urlB.lastIndexOf("/") + 1);
String text = req.getParameter (W_TEXT);
BufferedWriter buffer =
new BufferedWriter (new FileWriter (W_PAGE_DIR+page+".dot"));
buffer.write (text, 0, text.length ()); buffer.close ();
// chargement de la page sauvegardee
doGet (req, res);
}
catch (IOException e) { res.sendError (HttpServletResponse.
SC_SERVICE_UNAVAILABLE);
}
```

## Commentaires

- 1 méthode appelée lorsque le client demande l'URL via la méthode HTTP Post (après saisie du code wiki de la page dans un formulaire XHTML)
- 2 récupération du nom de la page
- 3 sauvegarde du code wiki dans le fichier correspondant
- 4 envoi du code XHTML correspondant au client, via la méthode doGet



# XHTMLBody

```
private String XHTMLBody()
```

```
if (W_action == null) { /* normal XHTML view */
return "<body><p><a href=\""+W_page+"?action="+ W_EDIT + "\">Edit</a><br
/>"
+wikiToHtml.parse(loadFile(W_PAGE_DIR+W_page+".dot"))+"</body>";};
if (W_action.equals(W_EDIT)){ /* edit XHTML view (form) */
String rep="<form method=\"post\" action=\""+W_page+"\">\n";
rep+="<p><textarea cols=\"120\" rows=\"40\" name=\""+ W_TEXT + "\">"
+ loadFile(W_PAGE_DIR+W_page+".dot")+"</textarea></p>\n";
rep+="<p><input type=\"submit\" value=\"Send\" /></p></form>";
return rep;}
return "";
```

## Commentaires

- 1 retourne le corps XHTML en fonction du mode enregistré dans W\_action
- 2 si pas de mode défini : mode normal -> traduction syntaxe wiki en code XHTML via WikiParser et ajout hyperlien pour édition
- 3 si mode édition : on insère dans un formulaire le contenu de la page (syntaxe wiki), via loadFile, à transmettre à la méthode doPost (pour enregistrement dans le fichier)
- 4 autres modes (print etc), à faire ...

```
private String loadFile( String file )
```

```
try{  
    BufferedReader buffer = new BufferedReader (new FileReader (file));  
    StringBuffer bufferString = new StringBuffer();  
    String line;  
    while ((line = buffer.readLine ()) != null)  
        bufferString.append (line+"\n");  
    return bufferString.toString();  
  
    catch (FileNotFoundException e) {  
        return loadFile(W_PAGE_DIR+"FileNotFoundException.dot");  
  
        catch (IOException e){  
            return loadFile(W_PAGE_DIR+"IOException.dot");  
        }  
    }  
}
```

## Commentaires

- 1 retourne le contenu du fichier texte `file`
- 2 si le fichier n'existe pas, renvoie la page wiki `FileNotFoundException`, qui informe l'utilisateur que la page n'existe pas et propose de l'éditer (i.e. création d'une nouvelle page)
- 3 si erreur de lecture, renvoie la page wiki `IOException`, qui informe l'utilisateur d'un problème

## Interface Parser

```
public String parse(String input) ;
```

## WikiParser implements Parser

**Classe interne** : Replacement implements Parser

**Variable** : List<Parser> parsers

**Méthodes** :

- public WikiParser()
- public String parse(String input)

## Commentaires

- 1 la classe interne `Replacement` prend en charge une méthode de substitution via sa méthode `parse`
- 2 le constructeur construit une collection d'instances de `Replacement`
- 3 la méthode `parse` applique toutes les méthodes de substitutions de la collection `parsers`



# Replacement

```
class Replacement implements Parser
```

```
private Pattern pattern; private String replacement;

public Replacement(String regexp, String replacement) {
    // pre-compile regular expressions
    pattern=Pattern.compile(regexp); this.replacement=replacement;
}

public String parse(String input) { // use matcher, then replace all
    Matcher matcher = pattern.matcher(input);
    return matcher.replaceAll(replacement);
}
```

## Commentaires

- 1 classe interne pour la prise en charge d'une substitution sémantique pour les occurrences spécifiées par une expression régulière
- 2 la variable `pattern` est l'expression régulière compilée (pour augmenter les performances)
- 3 la variable `replacement` contient la chaîne de substitution
- 4 la méthode `parse` prend en paramètre la chaîne à substituer et effectue toutes les substitutions

# Constructeur WikiParser

```
public WikiParser()
```

```
    pars=new ArrayList<Parser>();  
    // URL type one [[internal page]] : replace [[(1:filename).(2:ext)]]  
    pars.add(new Replacement("\\\\[\\[([\\w_-]+)\\\\]", "<a href=\"\\$1\\>\\$1</a>  
        >"));  
    // URL type two [[internal_page|name]] : replace by "<a href=\"page=  
        internal_page\\>name</a>"  
    pars.add(new Replacement("\\\\[\\[([\\w_-]+)\\\\|([\\w\\p{Punct}\\p{Blank  
        }]+)\\\\]", "<a href=\"\\$1\\>\\$2</a>"));  
    // line feed "\\n" x k and k >=2 -> <br />  
    pars.add(new Replacement("(\\n(?:\\p{Blank}{0,}){2,})", "<br />\\n"));  
    // list block ("* text" x n) parser adds <ul> ... </ul>  
    pars.add(new Replacement("(?:\\* [^\\n]+\\n)+", "<ul>\\n\\$1</ul>\\n"));  
    // list item ("* text") identifier adds <li> ... </li>  
    pars.add(new Replacement("\\* ([^\\n]+)\\n", "<li>\\$1</li>\\n"));
```

## Commentaires

- 1 dans une expression régulière, on utilise \ pour protéger un caractère
- 2 dans une chaîne, on utilise \\ pour le symbole \; \w désigne un caractère de mot
- 3 [[internalpage]] ↦ <a href="internalpage">internalpage</a>  
expression régulière : \\[\\[([\\w\_- +)\\\\]
- 4 rajouter vos propres expressions régulières pour enrichir la syntaxe du wiki

```
public String parse(String input)
```

```
String temp=input;  
for (Iterator<Parser> it=pars.iterator(); it.hasNext();) {  
temp=it.next().parse(temp);  
}  
return temp;
```

## Commentaires

- 1 applique toutes les méthodes de substitutions de la collection `pars` à la chaîne passée en paramètre