

MA-004-193-1

République Algérienne Démocratique Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Saad Dahlab, Blida

Faculté des Sciences

Département d'Informatique

Mémoire de fin d'étude en vue d'obtention du Master en Informatique

Option : Ingénierie du logiciel



THÈME :

# TRANSFORMATION D'UN SITE JOOMLA! 2.5 EN WORDPRESS 3.5

MA-004-193-1

Encadreur :

Mr. CHERIF-ZAHAR Amine

Présidente de jury :

Mme. BOUSTIA Narimène

Réalisé par :

Mr. BENKHADEM Rafik

Promotion 2012-2013





# Dédicace

*J'ai le grand plaisir de dédier ce modeste travail, fruit de longues années d'études à ceux qui m'ont donné la vie, Mes parents.*

*A Mes chers frères et sœurs.*

*A tous les membres de ma famille, petit et grands ;*

*A tous mes amis ;*

*A tous ceux et celles qui me sont si chers ;*

*A toute personne qui me connaisse ;*

*Je vous dédie ce travail.*

*RAFIK*

# Remerciement

*Je tiens en premier lieu à remercier le bon Dieu tout puissant pour m'avoir aidé et facilité la réalisation de ce travail.*

*Je remercie mon promoteur Mr. CHERIF-ZAHAR Amine, professeur au Département d'Informatique de l'Université Saad Dahlab de Blida de m'avoir orienté et mis à ma disposition tous les moyens nécessaires pour parfaire mon travail.*

*Mr. BENNOUAR, professeur au Département d'Informatique de l'Université Saad Dahlab de Blida pour sa vertueuse contribution et ses conseils de valeur.*

*Mes remerciements vont également aux membres du jury.*

*A tous mes enseignants au Département d'informatique de l'Université Saad Dahlab de Blida et toute ma promotion de Master 2.*

*Enfin, je remercie toute personne qui m'a encouragé et aidé de près ou de loin.*

# Résumé

Les Systèmes de Gestion de Contenu (SGC) ou Content Management Système (CMS) en anglais, sont des logiciels qui permettent de créer et d'administrer des sites web à distance et via un browser classique, que ce soit sur internet ou sur un réseau local. Un CMS a pour rôle de base de gérer les principales fonctionnalités d'un site web, et tout traitement sur un domaine spécial requière un CMS dédié, chaque CMS a des points forts et des points faibles, Notre travail consiste à concevoir et à réaliser un processus automatique de migration des données (articles, médias, utilisateurs ...) de la plate-forme Joomla ! 2.5 vers la plate-forme WordPress 3.5.

**Mots clés :** Joomla, plateforme, SGC, Wordpress, site, migration.

# Abstract

The Content Management Systems (CMS) is software that creates and manages websites at distance through a standard browser, whether on the Internet or a local network. CMS has a major role of managing the main functions of a web site, and any treatment of a Special area requires a dedicated CMS, each CMS has strong points and weak points, our work is to design and implement an automatic data migration process (articles, media users ...) of the Joomla platform! 2.5 to the Wordpress platform 3.5.

**Keywords:** Joomla, platform, CMS, Wordpress, site, migration.

# Table des Matières

**Avant-propos**

**Introduction**

**Partie I : Etat de l'art**

**Chapitre 1 : Système de gestion de contenu**

1.1. Définitions.....	4
1.2. Evolution de la gestion des sites web.....	5
1.3. Pourquoi les CMS.....	7
1.4. Fonctionnalités d'un CMS.....	8
1.5. Caractéristiques.....	9
1.6. Domaines d'application et types de CMS associés.....	10
1.7. Synthèse.....	11

**Chapitre 2 : Le parallèle entre Joomla ! et WordPress**

2.1. Les qualités d'un bon CMS.....	13
2.2. Comparaison.....	13
2.2.1. Joomla ! .....	13
2.2.2. Wordpress.....	14
2.3. Etude comparative.....	14
2.4. Conclusion.....	18

**Partie II : Analyse, conception et réalisation du système**

**Chapitre 3 : MVC Joomla et WordPress**

**Introduction**

3.1. Joomla ! .....	22
3.1.1. Structures et termes de Joomla ! .....	22
A. Frontend.....	22
B. Backend.....	22
C. Fichiers.....	23

D. Les éléments de Joomla ! .....	23
E. Types de contenus .....	26
F. Base de données .....	28
3.1.2. Architecture de Joomla .....	30
3.1.3. Modèle MVC Joomla .....	31
3.1.4. Connexion MVC .....	32
3.1.5. Conclusion .....	33
3.2. WordPress .....	34
3.2.1. Structure de WordPress .....	34
A. Frontend .....	34
B. Tableau de bord.....	34
C. Fichiers .....	36
D. Base de données.....	36
E. Les éléments de WordPress .....	38
3.2.2. Architecture de Wordpress .....	38
3.3. Conclusion du chapitre .....	39

## **Chapitre 4 : Conception du traducteur**

### **Introduction**

4.1. Approches de Migration .....	41
4.1.1. Approche de migration des données via base de données .....	41
4.1.2. Approche de migration des données via fichier XML .....	46

## **Chapitre 5 : Réalisation et test**

### **Introduction**

5.1. Installation et mise en œuvre de la migration via BDD.....	50
5.1.1. Installation du composant .....	50
5.1.2. Utilisation.....	51
5.1.3. Test.....	53
5.2. Installation et mise en œuvre de la migration via fichier XML.....	55
5.2.1. Installation du composant.....	55
5.2.2. Les étapes de l'exportation du fichier XML.....	55
5.2.3. Les étapes de l'importation du fichier XML.....	56

5.2.4. Test.....	57
5.3. Discussion.....	59
5.4. Conclusion du chapitre.....	59

**Conclusion**

**Références Bibliographiques**



# Liste des Tableaux

<b>Tableau 1 : Spécificités techniques entre Joomla et WordPress.....</b>	<b>15</b>
<b>Tableau 2 : Spécificités fonctionnelles et extensions.....</b>	<b>15</b>
<b>Tableau 3 : Ergonomie, interface et facilité de prise en mains.....</b>	<b>16</b>
<b>Tableau 4 : Administration et sécurité.....</b>	<b>16</b>
<b>Tableau 5 : Récapitulatif.....</b>	<b>17</b>

# Liste des Figures

<b>Figure 1:</b> Schéma fonctionnel d'un site web statique.....	6
<b>Figure 2:</b> Schéma fonctionnel d'un site web dynamique.....	7
<b>Figure 3 :</b> Interface de l'Administrateur du CMS Joomla.....	8
<b>Figure 4 :</b> Joomla Administration registration.....	22
<b>Figure 5 :</b> Administration.....	22
<b>Figure 6 :</b> Articles en pages d'accueil.....	23
<b>Figure 7 :</b> Affichage d'un article seul.....	24
<b>Figure 8 :</b> Affectation des catégories.....	24
<b>Figure 9 :</b> Administration.....	26
<b>Figure 10 :</b> Formulaire d'article.....	27
<b>Figure 11 :</b> Modèle conceptuel de données « Joomla ! 2.5 » .....	29
<b>Figure 12 :</b> Joomla, architecture en couche.....	30
<b>Figure 13 :</b> Joomla, architecture en arbre.....	31
<b>Figure 14 :</b> Les modèles MVC de Joomla.....	32
<b>Figure 15 :</b> Connexion.....	34
<b>Figure 16 :</b> Tableau de bord (1) .....	35
<b>Figure 17 :</b> Tableau de bord (2) .....	36
<b>Figure 18 :</b> Modèle conceptuel de données « Wordpress 3.5 » .....	37
<b>Figure 19 :</b> Diagramme de classes partial « articles » .....	42
<b>Figure 20 :</b> Diagramme de classes partial « posts » .....	43
<b>Figure 21 :</b> Exemple de mappage des données.....	44
<b>Figure 22 :</b> Article à exporter.....	45
<b>Figure 23 :</b> Gestion des extensions.....	50
<b>Figure 24 :</b> Installeur d'extensions de Joomla.....	50
<b>Figure 25 :</b> Choix de l'extension à installer.....	51

<b>Figure 26</b> : Transfert de fichier et installation.....	51
<b>Figure 27</b> : Accès au composant.....	52
<b>Figure 28</b> : Ecran d'accueil du composant.....	52
<b>Figure 29</b> : Informations sur la BDD de destination.....	52
<b>Figure 30</b> : Les données migrées.....	53
<b>Figure 31</b> : Gestionnaire d'articles Joomla.....	53
<b>Figure 32</b> : Gestionnaire d'articles Wordpress.....	54
<b>Figure 33</b> : Article affiché par Joomla ! .....	54
<b>Figure 34</b> : Article migré de Joomla vers Wordpress.....	55
<b>Figure 35</b> : Génération du fichier XML.....	55
<b>Figure 36</b> : Accéder au menu outils.....	56
<b>Figure 37</b> : Sélection du plugin « importer » .....	56
<b>Figure 38</b> : Sélection du fichier d'exportation.....	57
<b>Figure 39</b> : Sélection des options d'exportation.....	57
<b>Figure 40</b> : Article affiché par la plateforme Joomla ! .....	58
<b>Figure 41</b> : Résultat d'exportation.....	58

## Introduction

Le monde de l'informatique a connu un nouvel essor à la fin des années 90, avec la généralisation de l'Internet qui a permis de communiquer, d'échanger des documents et autres fichiers et surtout de mettre au point plus facilement des réseaux VPN distants pour les entreprises. Ceci devint indispensable ces dernières années avec l'explosion du haut débit, surtout pour les entreprises.

Pour une plus grande rapidité, la gestion des documents et des fichiers, qui se faisait par programmes classiques après qu'ils soient reçus via le réseau, doit se faire directement via un outil Framework qui est le système de gestion de contenu web.

Les systèmes de gestion de contenu (S.G.C) pour Content Management System(C.M.S) en anglais, sont des outils de gestion dynamique de sites web et d'applications multimédias de manière interactive à l'aide de divers modules. Ils permettent à plusieurs personnes selon des droits prédéfinis de gérer leur site web localement ou à distance, en leur fournissant une chaîne de publication (Workflow) pour la mise en ligne, la mise à jour, et la suppression de documents. Tous cela en assurant une indépendance entre les opérations de gestion de la forme et du contenu.

Les CMS existent depuis les années 80, mais n'ont pas connu de grands succès qu'avec leur utilisation dans le web. Leur fonction de base étant de structurer le contenu afin de faciliter la gestion des services tels que les forums, FAQ ... etc.

Pour notre part nous nous intéressons aux deux systèmes de gestion de contenu en particulier qui vont être le sujet de notre étude, Notre application de migration doit ainsi gérer en particulier les éléments suivants :

- migration des données (articles, pages statiques ...etc.).
- transfère des fichiers et médias du répertoire de Joomla ! vers les répertoires du Wordpress.
- modification des liens internes du contenu, et garder les droits d'accès.

L'objectif est de faciliter la migration de Joomla vers WordPress, qui a empêché plusieurs développeurs web de le faire puisque ce processus requiert trop de travail manuel.

Afin d'atteindre les objectifs planifiés, notre document est organisé de la manière suivante :

- La première partie sert à décrire les systèmes de gestion de contenu en général, et ceux qui constituent le thème du projet.
- Le chapitre 1 décrit les CMS et donne un aperçu de leur emploi, leurs fonctionnalités et leur classification et met en valeur la nécessité des CMS.
- Le chapitre 2 présente les deux CMS (Joomla et WordPress) avec une étude comparative.
- Le chapitre 3 décrit le paradigme de programmation « MVC », structure et architecture de Joomla ! et WordPress.
- Le chapitre 4 montre la conception du traducteur.
- Le chapitre 5 montre les résultats et tests de ce travail.

# PARTIE I :

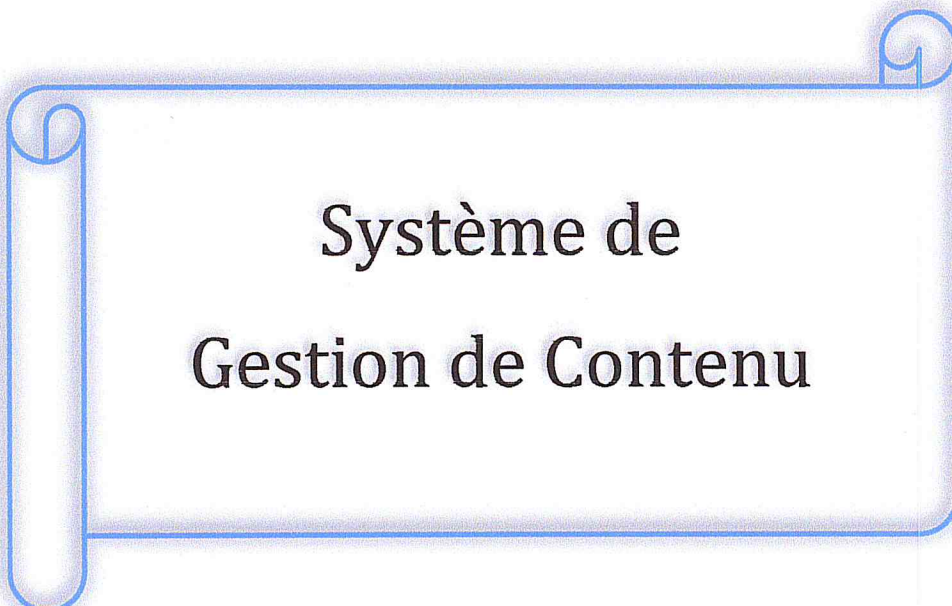
## Etat De L'Art

### **Objectif :**

Dans cette partie nous allons introduire les concepts de base des CMS, leurs caractéristiques générales et leur évolution.

Nous allons ensuite présenter les CMS Joomla et WordPress qui font l'objet de notre travail avec une étude comparative des différents outils existants.

# CHAPITRE 1



Système de  
Gestion de Contenu

« La connaissance des mots conduit à la connaissance des choses. »

Platon

Le but de cette partie est de bien connaître ce que c'est un système de gestion de contenu, ce qui est nécessaire avant d'entamer notre projet.

### 1.1. Définitions

Afin de bien cerner le concept des CMS pour Content Management System qui est la traduction anglaise de Système de Gestion de Contenu (SGC), nous donnons dans cette partie quelques définitions dont la plupart tirée de la littérature.

#### Définition 1

Une première définition présente les Systèmes de Gestion de Contenu (Content Management Systems -CMS) comme étant *des outils qui permettent de publier et d'actualiser les contenus d'un site internet, généralement via de simples formulaires. Les informations encodées par l'utilisateur sont alors automatiquement présentées d'une façon structurée et cohérente sur le site, sans que celui-ci ne doit maîtriser le code HTML, normalement nécessaire à la mise en forme de textes et d'images pour le web.*

*Les CMS sont souvent conçus pour permettre un travail collaboratif entre utilisateurs [WEB\_01].*

#### Définition 2

[WEB\_02] propose une autre définition pour les CMS comme suit :

*Système intégré et multi plate-forme permettant aux entreprises et aux organisations de gérer rapidement et facilement le contenu dynamique et rédactionnel d'un site Web. Les systèmes de gestion de contenu sont des solutions flexibles et dynamiques qui permettent aux sociétés une actualisation facile des sites Web, c'est-à-dire de corriger et d'ajouter des textes, des photos et des fonctions multimédias sur leur site, sans avoir besoin d'aide extérieure.*

#### Définition 3

[WEB\_03] quant à lui définit les CMS Comme suit :

*Les systèmes de gestion de contenu sont devenus incontournables pour créer et mettre en ligne du contenu sur internet et les intranets. Ces logiciels, souvent orientés gestion et partage d'information, gère le cycle de vie de l'information depuis sa création avec les éditeurs, en passant par l'organisation des ressources, jusqu'au déploiement en ligne lors de la publication, et enfin à l'archivage des éléments de contenu. Ces applications ont pour caractéristique de séparer le contenu de la forme ce*

*qui permet de traiter indépendamment le contenu de son aspect final à l'écran. L'avantage de cette méthode est de faciliter la maintenance d'un site et de permettre un partage efficace du travail.*

Pour notre part un système de gestion de contenu est :

«Un logiciel fonctionnant en réseau (internet ou intranet) qui gère les différentes fonctionnalités d'un site web dynamiquement. Le CMS sépare le contenu de la présentation et offre à l'utilisateur une interface simple et conviviale, incluant un ensemble de modules personnalisables, qui lui évite toute manœuvre de programmation nécessaire à la conception du site web que ce soit pour la présentation (HTML), ou la gestion du contenu (PHP, ASP, ...). Un avantage certain pour le CMS est la possibilité d'un travail collaboratif à distance et la facilité de la maintenance du site ainsi que sa mise à jour. »

### 1.2. Evolution de la gestion des sites web

Le début des années 1990 est marqué par l'expansion du web, à cette époque les sites web étaient développés directement en manipulant le langage HTML, de manière statique (voir Figure 1 : Schéma fonctionnel d'un site web statique, par des développeurs, qui dans la majorité des cas ne maîtrisaient pas bien le langage, avec de simples éditeurs de texte ou des logiciels pas assez perfectionnés, cela donnait souvent des résultats très satisfaisants pour la création surtout au niveau esthétique. Mais si la création a permis à plusieurs développeurs novices ou experts de s'affirmer, ce n'était pas le cas pour la maintenance qui ne tardait pas à être nécessaire, avec ces sites statiques où seuls de rares développeurs très expérimentés pouvait minimiser le temps de la maintenance; et pour des raisons de coût, il n'y avait qu'une seule personne (le webmaster) qui entretenait le site. Le web se généralisa et le besoin en diffusion d'informations et d'interaction avec les utilisateurs augmenta de façon exponentielle, le webmaster ne pouvant supporter la maintenance des sites à lui seul, il est devenu un goulot d'étranglement, les sites statiques ne répondaient plus aux besoins de presque tous les utilisateurs (entreprises, internautes, journaux ...). Pour cela, est apparue la notion de sites dynamiques pour faciliter la mise à jour et la maintenance, en organisant les données dans une base de données indépendamment de leur présentation avec une génération de pages web à la demande en séparant le contenu (l'information) du contenant (la mise en page) (voir Figure 2).



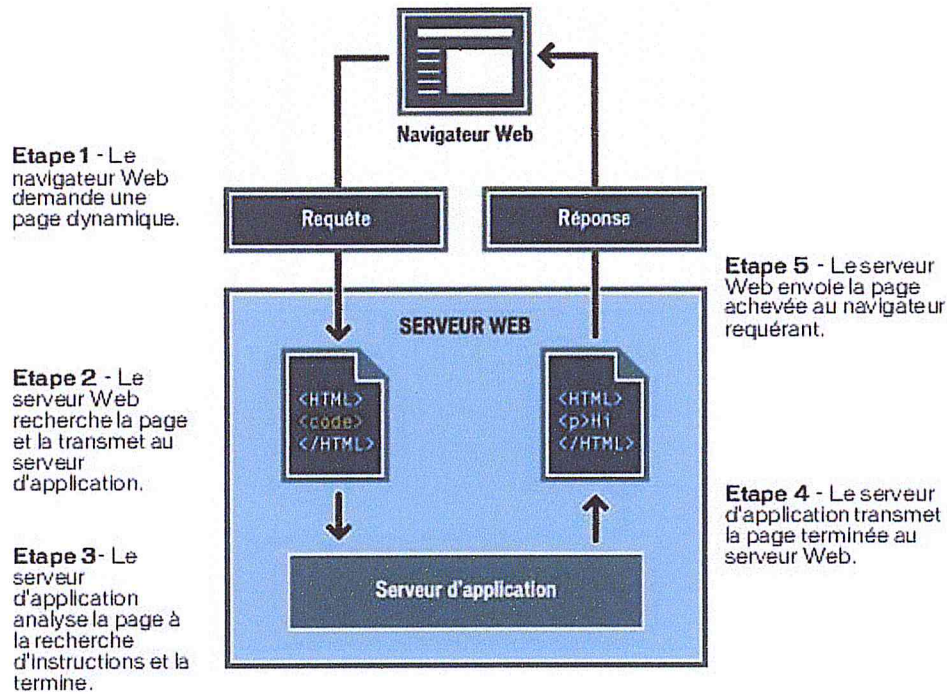


Figure 1 : Schéma fonctionnel d'un site web statique. [WEB\_04]

Une première solution pour séparer le contenu du contenant est apparue avec l'apparition des langages orientés web et des scripts CGI, le contenu étant stocké dans des bases de données celle-ci est reliée aux pages à l'aide de scripts écrits dans un langage de programmation tels que PHP, PERL .... Ceci a permis de régler les problèmes de maintenance, et les nouveaux contenus étant écrits directement et sauvegardés dans la base, l'affichage se fait automatiquement à l'aide des scripts qui se chargent de la mise en page.

Cette approche comporte quelques inconvénients parmi lesquels : les scripts ne sont valables que pour un certain style de site, si on voulait changer de style ou bien ajouter des fonctionnalités pour le site il fallait changer les scripts ou partie d'entre eux, on créait d'autres adaptés à ceux qui existaient ou le contraire avec les coûts que cela engendre, ceci est dû au fait que les bases de données relationnelles ne sont pas adaptées à ce genre de documents (les bases de données objet parues récemment le sont plus). Dès 1995, la deuxième solution qui est venue améliorer la première a commencé à voir le jour, l'idée était de créer carrément un logiciel accessible par tous les utilisateurs et qui gère automatiquement tous les aspects liés à la fusion du contenu au contenant. En 1997, on a commencé à les concevoir par des développeurs isolés seuls ou dans des entreprises, en 1999 les premiers CMS qui ont vu le jour étaient essentiellement des CMS propriétaires et assez chers. En 2002, les CMS ont connu un nouvel essor par leur ouverture à l'open source (comme Joomla, SPIP, Drupal), ce qui a

chamboulé la manière de se comporter avec la création d'applications web, qui est devenu accessible à tout le monde.

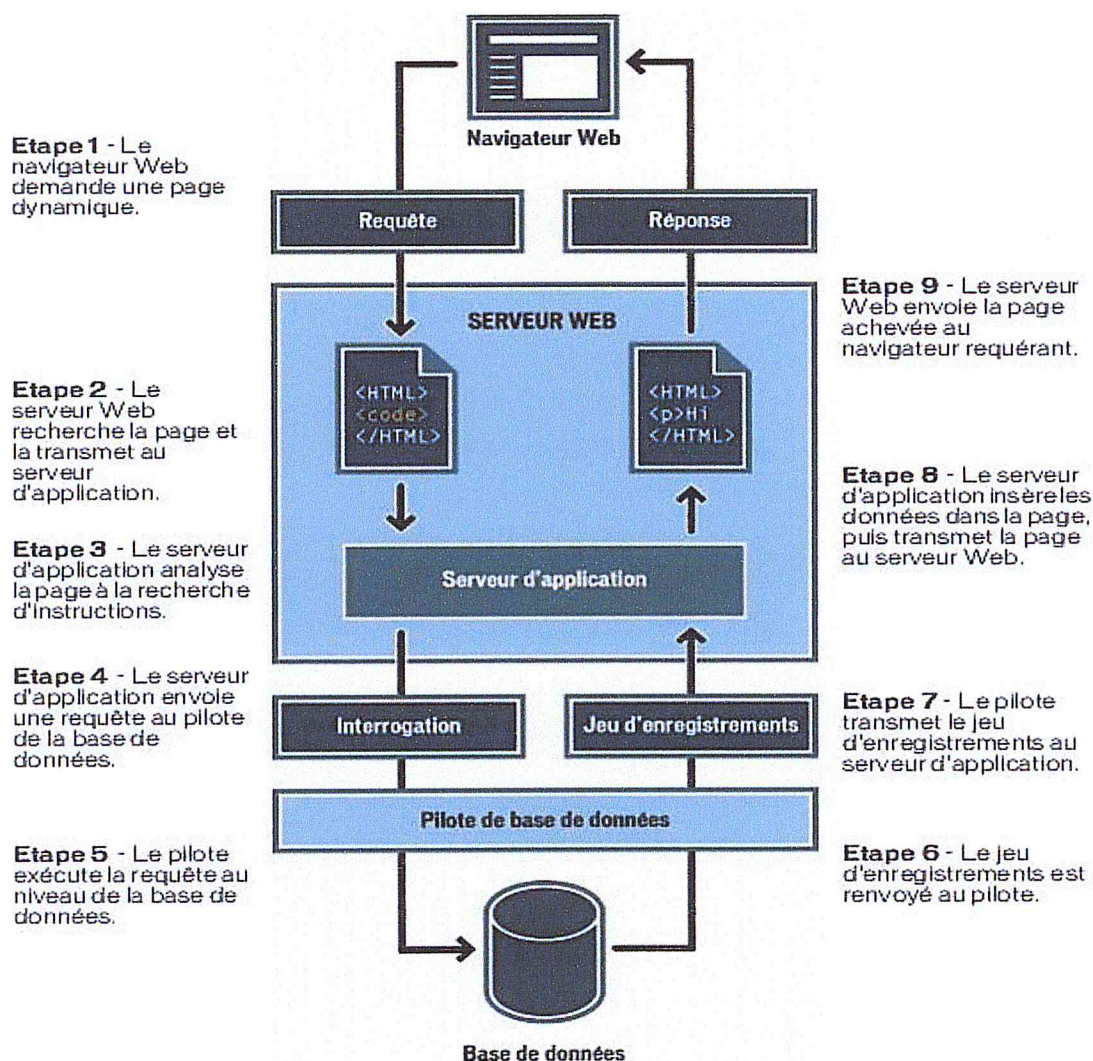


Figure 2: Schéma fonctionnel d'un site web dynamique. [WEB\_05]

Depuis, les CMS ne cessent de s'améliorer, offrant de plus en plus de fonctionnalités et plus de facilité de gestion des sites, le besoin d'un webmaster qui gère tout n'est plus d'actualité et le besoin d'informaticien spécialiste dans le domaine n'est plus indispensable dans la majorité des cas.

### 1.3. Pourquoi les CMS

Le web n'est plus un outil de luxe pour les particuliers, ni une simple interface pour mettre en valeur ses produits par des publicités. Le haut débit est plus facile d'accès d'où la possibilité de baser son commerce sur le net ou bien gérer des informations en temps réel, d'organiser des forums à très long terme, de créer ses propres blogs, d'effectuer ses achats en ligne et mettre en œuvre des wikis... Ceci

implique que utilisation d'un webmaster ou même une application classique au niveau serveur constituera un goulot d'étranglement vu le débit élevé de la connexion internet de nos jours.

Les CMS sont venus régler ce problème, cela a permis de gérer directement à partir d'un simple navigateur web les sites publiés à partir de n'importe quel point du réseau, selon des droits prédéfinis. Les modifications peuvent être instantanées (poster un commentaire pour un article ou un forum...) ou bien si cela nécessite comme même une vérification d'un administrateur (article publié par un journaliste...), il n'aura qu'à valider l'article pour qu'il s'affiche instantanément.

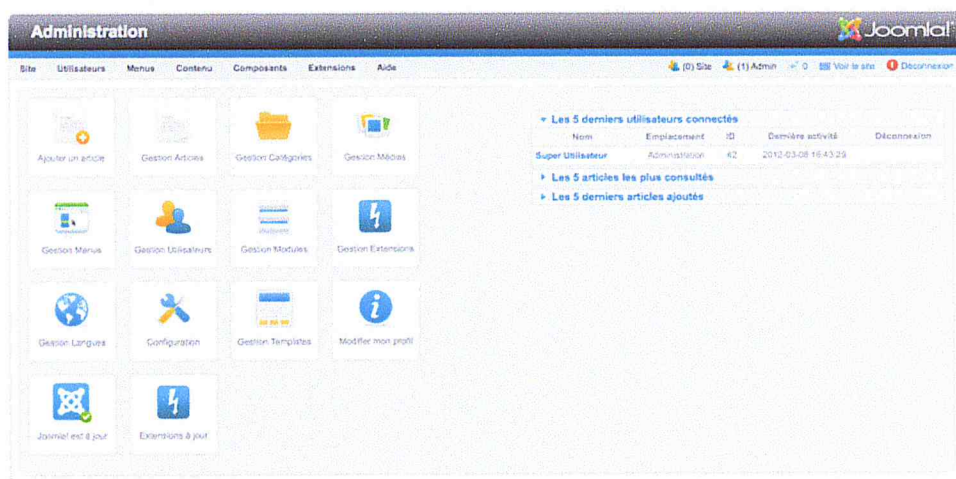


Figure 3: Interface de l'Administrateur du CMS Joomla

Pour satisfaire le besoin de rapidité des entreprises qui veulent doubler leurs concurrents ou tout simplement l'impatience de n'importe quel utilisateur à voir s'afficher son travail rapidement, les CMS offrent une solution idéale, très simple à utiliser, on peut même devenir maître dans la création de contenu web avec une bonne formation sur un CMS donné.

Bien sûr, les CMS ne règlent pas tous les problèmes liés à la sécurité, aux réseaux ....

Des spécialistes dans le domaine restent indispensables pour certains projets d'envergure, et les CMS sont en constante évolution pour répondre de plus en plus à des besoins spécifiques et non plus à des besoins généraux.

### 1.4. Fonctionnalités d'un CMS

Parmi les fonctionnalités requises dans un système de gestion de contenu nous citons :

### • Gestion du contenu

- Gestion des documents structurés (fournis par une base de données ou des fichiers XML) ou non structurés (fichiers HTML, PDF, WORD ...) ainsi que les versions, l'archive, historique, les formats, les mises à jour ;
- Utilisation des métadonnées ;
- Gestion des pages et des rubriques ;
- Gestion automatisé des liens amont et aval, intra et extra document.

### • Gestion des utilisateurs et des droits

- Travail collaboratif, cela permet la décentralisation et être le plus loin des goulots d'étranglement liés à la dépendance du webmaster ;
- Gestion des utilisateurs, des profils et des droits d'accès en référence aux documents ou aux actions liés à eux.

### • Processus de publication

- Présence de modules supplémentaires qui permettent d'enrichir le CMS avec de nouvelles fonctions pouvant être personnalisés selon les besoins de l'utilisateur ;
- Une communication basée sur un système de validation des documents (cycle de vie d'un document)
- Différents types de diffusion (selon le CMS).

### • Fonctions supplémentaires

- Intégration de données externes : bases de données, annuaires, fichiers XML... ;
- Fonctions de recherche, des forums, des FAQs, aide contextuelle, sondages, chat, questionnaires Statistiques, Gestion des versions etc. ;
- Présence de modèles (en anglais templates), de gabarits (thèmes) pour la présentation du site.

### • Sécurité et confidentialité

- Maintien et maîtrise de la sécurité des comptes des utilisateurs ;
- Administration et configuration des droits ;
- Sauvegarde de copie et de l'historique des modifications.

## 1.5. Caractéristiques

Parmi les principales caractéristiques d'un CMS on peut citer:

- Facile à utiliser, ne requière pas de connaissances particulières tant pour les concepteurs du site que pour les rédacteurs et les lecteurs ;

- Accès depuis un navigateur web via le réseau (internet, intranet, extranet) depuis n'importe quel point de celui-ci et avec n'importe quel système d'exploitation ;
- Création et publication dans un FrontOffice (qui est le site web) à travers un backoffice fourni à chaque utilisateur selon ses droits ;
- Séparation du contenu et du contenant ;
- Propose une gestion de cycle de vie des documents (Workflow) ;
- Plusieurs personnes peuvent modifier le contenu selon leurs droits et leur champ d'application (Travail collaboratif) ;
- Possibilité de personnaliser un CMS (open source) pour ceux qui maîtrisent le langage utilisé pour celui-ci.
- Modulable, possibilité d'ajouter des fonctionnalités selon le besoin par des plugins, extensions... surtout avec l'apparition des CMS open source, les CMS propriétaires devaient suivre en proposant des modules complexes en plus payant ;
- Ergonomique, un CMS doit donner une souplesse aux utilisateurs, il doit générer des pages sécurisées, lisibles, accessibles.... ;
- Conviviale, Ce ne doit vraiment pas être nécessaire d'avoir les compétences d'un développeur web ou un spécialiste pour utiliser un CMS.

### 1.6. Domaines d'application et types de CMS associés

Les CMS sont utilisés dans plusieurs domaines qui ont un besoin spécifique d'où l'apparition de plusieurs types de CMS...

#### 1.6.1. Domaines d'application

Les CMS en plus de pouvoir créer facilement des pages web dynamiques, sont devenus l'ultime besoin de toute institution, entreprise ou individus désirants publier du contenu sur le web, c'est pour cela qu'ils présentent l'outil le plus utilisé en ce moment sur Internet, et qui a permis une meilleure flexibilité de publication. En effet, il existe plusieurs domaines qui font appel à ce dernier vu la diversité des utilisateurs ainsi que leur emploi.

Parmi ceux-ci citons :

**Les sites institutionnels** : C'est le type de site le plus répandu sur le web. Le site institutionnel est utilisé par toutes les entreprises et administrations qui souhaitent être présentes sur Internet. Le visiteur y trouve toutes les informations concernant l'organisation, les activités, les services, les produits, etc.

**Les sites éditoriaux :** Comme pour l'institutionnel, le site éditorial est très répandu. Il est utilisé par des groupes d'individus ou des entités qui souhaitent se positionner comme source d'information ou de veille. Typiquement, les journaux en ligne et les agences de presse utilisent ce type de site.

**Les communautés en ligne :** Le site de communauté en ligne regroupe des internautes qui ont un centre d'intérêt commun. C'est un lieu d'échange et de diffusion du savoir organisé autour d'un thème central suffisamment fédérateur. Il propose un contenu et des services pour les membres de la communauté. Le module principal servant aux échanges est généralement un forum modéré ou non, qui peut être décliné en thèmes et sous thèmes.

**Les sites catalogues et marchands :** Le site catalogue présente une gamme de services, de produits et essaie de créer un contexte favorable pour que l'internaute choisisse l'entreprise plutôt qu'une autre. L'étape suivante est le site marchand qui offre, en plus, des possibilités de commande et de paiement en ligne. Ce type de site demande souvent la mise en place d'une logistique importante pour la gestion des stocks et des expéditions.

**Les Intranet/Extranet :** C'est l'une des applications en plein essor aujourd'hui. En effet, un outil de gestion de contenu offre des fonctionnalités qui favorisent les échanges entre les salariés au sein d'une entreprise, entre une entreprise et ses clients/fournisseurs et entre une administration et ses administrés. Ce type de site offre un contenu à forte valeur ajoutée et des fonctionnalités de travail collaboratif. Nous allons y trouver des espaces privés favorisant la relation avec des visiteurs identifiés, des espaces de collaboration, des inscriptions à des listes de diffusion qui permettront de recevoir des lettres d'informations... [BAD\_01]

### 1.7. Synthèse

A partir des éléments précédents, nous donnons dans ce qui suit la synthèse suivante

#### • Avantages des CMS

- ✂ Séparation du contenu du contenant ;
- ✂ Décentralisation de la gestion de contenu ;
- ✂ Facilité, souplesse, ergonomie... ;
- ✂ Rapidité de mise en œuvre de pages web.

#### • Inconvénients des CMS

- ✂ Propose trop de fonctionnalités souvent inutiles pour des besoins spécifiques ;
- ✂ Par contre, pour ces mêmes besoins il n'existe pas assez de fonctionnalités.

# CHAPITRE 2

Le Parallèle  
Entre  
Joomla et Wordpress

Le but de ce chapitre est de faire une comparaison entre les fonctionnalités de JOOMLA ! et de WordPress.

### 2.1. Les qualités d'un bon CMS :

Un bon CMS doit être :

- facile à utiliser mais extensible selon les besoins : dans la pratique ces deux caractéristiques sont souvent incompatibles.
- prêt à l'emploi et modulable à volonté : la composition d'un CMS doit en faire un outil très performant dans la durée. Si le CMS est modulable, alors les options se résumeront au strict minimum au départ mais une bibliothèque fournie de modules additionnels offrira le maximum de flexibilité au fil du temps.
- équilibré entre options natives et modules additionnels.
- Ces notions sont évidemment partiellement incompatibles : les CMS les plus faciles à utiliser pour un débutant sont les plus simples et souvent les moins évolutifs dans la durée tandis que les solutions les plus complètes du marché sont logiquement les moins faciles à prendre en main.

### 2.2. Comparaison :

#### 2.2.1. JOOMLA :

##### Les atouts de Joomla :

- Facile à installer
- Configuration assez poussée possible
- Bon équilibre entre les options natives et plug-ins disponibles
- Nombreux templates sont disponibles pour débiter avec Joomla

Les novices et les développeurs en herbe seront ravis d'utiliser **Joomla** pour débiter. Mais **Joomla** montre ses limites si on décide de pousser très loin la personnalisation. Une refonte complète de **Joomla** s'avère presque impossible du fait de l'interconnexion très poussée des différentes briques fonctionnelles entre elles.

Là où Joomla devient une usine à gaz, c'est lorsqu'on le surcharge de modules. Particulièrement au début lorsqu'on teste et cherche le bon plugin ; la bibliothèque contient de nombreux plugins mais pas tous « maintenus à jour » et nombre d'entre eux ne fonctionnent pas correctement. [WEB06]



### 2.2.2 WORDPRESS:

WordPress est à n'en pas douter le CMS le plus couramment utilisé de nos jours. Il est principalement connu et reconnu pour ses **fonctions de blogging**, où il excelle littéralement. J'en sais quelque chose. Pour autant, WordPress n'est pas un « vrai CMS » selon moi, moins que Joomla ou Drupal en tous les cas. En revanche, **WordPress** est très puissant et possède un avantage indéniable grâce à sa bibliothèque d'extensions, on peut tout faire avec WordPress : *boutique en ligne, réseau social, multi-blogs, etc.* De même que pour Joomla, gare à ne pas le rendre ingérable si en le surchargeant de modules. De plus, les mises-à-jour intempestives (elles sont fort heureusement légères) font courir un (*faible*) risque d'incompatibilité entre la version installée et les modules (développées pour des versions antérieures).

Au final **WordPress** s'impose haut la main dans le cadre de projets légers, statiques ou contributifs. Les webzines et les blogs sont la chasse gardée de WordPress. Pour les projets plus lourds et plus ambitieux visant des sites de contenu profonds, envisagez Drupal ou Joomla à la rigueur.

### 2.3. Etude comparative <sup>1</sup> :

- WordPress marque des points dans l'ergonomie et la facilité de prise en mains, mais Joomla se distingue par son socle technique performant et ses possibilités de gestion multisites.
- WordPress est Gratuit: c'est le **leader des CMS Open Source**. Il est utilisé par des milliers d'internautes et développeurs partout dans le monde. Sa gratuité permet de réduire considérablement les coûts de conception
- A l'origine positionné en tant que "simple" moteur de blogs, WordPress a montré depuis toute sa pertinence pour également être utilisé pour gérer des sites Web complets. Une explication à cela : la qualité de son interface et son aspect "user-friendly". Aujourd'hui, des sites comme TechCrunch, Levi's ou encore Pepsi l'utilisent. De son côté, Joomla est considéré comme une solution de gestion de contenu Open Source, très appréciée des développeurs en particulier pour son aspect "boîte à outil" et la possibilité d'aller chatouiller le code de très près. Orange, IKEA mais aussi eBay l'ont choisi.
- Joomla est un CMS bâti sur un framework orienté objet: Techniquement, la grande différence entre WordPress et Joomla se situe certainement au niveau du "style" de code. WordPress est plutôt en PHP procédural, même s'il est possible d'utiliser des objets dans son code, alors que Joomla propose un framework orienté objet. Même si cette distinction n'a que peu d'impact sur l'utilisateur final, elle est importante pour le développeur qui doit aller voir sous le capot.

---

<sup>1</sup> Paragraphe pris du : « CMS Open Source : le match WordPress contre Joomla » Filippone.D, 2012

- Les développeurs apprécieront donc tout particulièrement l'ouverture complète du code de Joomla et les possibilités de le manipuler et d'un point de vue technique, Joomla est conçu comme un socle MVC très bien maîtrisé par les développeurs alors que WordPress repose sur des mécanismes de hook, avec des rajouts d'action, qui satisferont les utilisateurs non développeurs (Tableau 1).

Spécificités techniques		
	Joomla	Wordpress
Framework orienté objet	+++	
Faisceaux de sites multilingues	+++	+
Workflow nativement intégré	+++	
Ecriture des propres scripts de reprise de contenus	+++	+++
API XML-RPC et connecteur LDAP	+++	+
Gestion de la performance et cache de fichiers	+++	+
Source : JDN Solutions		

**Tableau 1 : Spécificités techniques entre Joomla et WordPress**

- E-commerce et groupware sont des points forts pour Joomla : Chez Joomla, on notera la présence d'un module e-commerce virtuel beaucoup plus avancé que celui de WordPress, de même qu'un module de groupware comprenant messagerie, contacts, annuaires et calendriers, beaucoup plus mature (Tableau 2).

Spécificités fonctionnelles et extensions		
	Joomla	WordPress
E-commerce	+++	+
Groupware	+++	+
Nombre d'extensions	++	+++
Source : JDN Solutions		

**Tableau 2 : Spécificités fonctionnelles et extensions entre Joomla et WordPress**

- WordPress a une longueur d'avance sur l'interface et l'ergonomie: Parmi les atouts affichés par WordPress, la qualité de l'interface utilisateur, celle de la création de thèmes, et son design. WordPress est un système très simple à utiliser, agréable à l'œil et permet de publier rapidement des contenus. Et même si les deux CMS font partie des plus abordables, WordPress est champion toute

catégorie dans ce domaine. Depuis sa conception et à chaque version, une grande attention est portée au design et à l'ergonomie de l'administration (Tableau 3).

Ergonomie, interface et facilité de prise en mains		
	Joomla	WordPress
Création de thèmes	+	+++
Attractivité et design	+	+++
Qualité de l'ergonomie	++	+++
Solution prête à l'emploi ne nécessitant aucune connaissance technique	+	+++
Source : JDN Solutions		

**Tableau 3 : Ergonomie, interface et facilité de prise en mains**

- WordPress est modulable et respecte les standards du web: il bénéficie d'une grande souplesse grâce aux très **nombreux plugins** disponibles via sa communauté. permet l'interopérabilité entre les différents navigateurs et la compatibilité avec les outils de lecture (Tableau 4).

- En termes de sécurité : Joomla est d'avantage centré sur la fonction éditoriale et l'organisation des contenus, ce qui n'est pas le cas de WordPress, utilisé à l'origine pour la publication de posts sur des blogs. Joomla propose une meilleure organisation hiérarchique des contenus (Tableau 4).

Dans leur version à jour, ces deux CMS, sont équivalents en termes de sécurité. Toutefois, WordPress est aujourd'hui le CMS le plus utilisé au monde, et fait donc logiquement partie des cibles privilégiées de ceux qui recherchent et exploitent des failles de sécurité. La communauté étant cependant nombreuse et réactive, des mises à jour sortent très souvent pour combler les failles détectées, et, dans la grande majorité des cas, avant qu'elles soient publiques. Mais il peut être difficile pour un webmaster de suivre le rythme de ces mises à jour (Tableau 4).

Administration et sécurité		
	Joomla	WordPress
Organisation hiérarchique des contenus	+++	++
Logique de droits associés aux contenus	+++	++
Faiblesse de l'exposition au risque de faille	++	+++
Source : JDN Solutions		

**Tableau 4 : Administration et sécurité**

- WordPress est idéal pour un site vitrine et Joomla pour une usine à sites: Outre une communauté plus importante, WordPress peut compter sur un catalogue d'extensions particulièrement dense pour

séduire les organisations publiques et privées. Ses points fort: la qualité de son interface, sa facilité de prise en mains par des profils d'utilisateurs non techniciens et son côté "prêt à l'emploi".

De son côté, Joomla apparaît comme un outil présentant des caractéristiques techniques poussées à même de séduire les développeurs souhaitant atteindre un très haut niveau de personnalisation. Dotée de solides fonctions e-commerce et de groupware, la solution se démarque aussi par ses performances, en termes de disponibilité grâce à son cache de fichiers.

Une entreprise qui penche pour la création d'une véritable usine à sites à tout intérêt à s'orienter vers Joomla, alors que celle qui souhaite d'abord se concentrer sur la création d'un site vitrine particulièrement esthétique choisira sans hésitation WordPress.

Tableau de synthèse du comparatif Joomla / WordPress		
	Joomla	WordPress
Taille de la communauté	++	+++
Coûts annexes réduits	+	+++
Framework orienté objet	+++	
Faisceaux de sites multilingues	+++	+
Workflow nativement intégré	+++	
Ecriture des propres scripts de reprise de contenus	+++	+++
API XML-RPC et connecteur LDAP	+++	+
Gestion de la performance et cache de fichiers	+++	+
E-commerce	+++	+
Groupware	+++	+
Nombre d'extensions	++	+++
Création de thèmes	+	+++
Attractivité et design	+	+++
Qualité de l'ergonomie	+	+++
Solution prête à l'emploi ne nécessitant aucune connaissance technique	+	+++
Organisation hiérarchique des contenus	+++	++
Logique de droits associés aux contenus	+++	++
Faiblesse de l'exposition au risque de faille	++	+++
Total	41 / 54	36 / 54
Source : JDN Solutions		

**Tableau 5: Récapitulatif**

- Enfin dans le critère de référencement, la première place est donnée à WordPress. Joomla! possède une option pour les URLs explicites, mais il affiche le chemin absolu de l'article (par exemple : *site.com/categorie1/categorie2/article*) tandis que WordPress va afficher directement l'article.

#### **2.4. Conclusion de l'étude comparative:**

D'origine destinée à la conception de blog, WordPress est devenu l'outil incontournable dans la liste des systèmes de gestion de contenus. Considéré comme le CMS le plus simple d'utilisation, il est conseillé pour les débutants désireux de se lancer dans la création de leur site. Simple et intuitif, il permet d'effectuer un panel conséquent d'actions, et ce, notamment grâce aux mises à jour régulières, il est réputé pour être l'un des CMS les plus simples d'utilisation.

# **PARTIE II :**

## **Analyse, Conception et Réalisation du système**

### **Objectif**

Dans cette partie nous allons étudier les besoins pour lesquels notre application de migration devra répondre, Par la suite, nous définissons les principaux axes de la conception et mise au point de l'application, cela englobe une présentation du système à l'aide d'UML et les différentes étapes de la réalisation du système pour finir par les tests et résultats obtenus.

# CHAPITRE 3

MVC

Joomla

Et Wordpress

### **Introduction :**

Pour pouvoir envisager une traduction d'une plate-forme à l'autre, il faut comprendre la structure de la base de données de chacune des plateformes, la hiérarchies des objets et les relations qui existent entre eux (composants, modules, articles, catégories, menus, templates etc.) et le modèle ayant servi au développement de chacune des plateformes (MVC ...).

### **Les modèles conceptuels des données des bases de données**

Pour pouvoir remonter aux schémas de base de données de chaque plate-forme on est besoin des outils de 'reverse engineering.

### **Reverse engineering**

Le reverse engineering est le processus qui consiste à générer un schéma de BDD à partir d'une structure de BDD existante. Pour faire faire on a utilisé un outil Visual « MySQL workbench ». [wiki\_01]

### **MySQL workbench**

Il permet à un administrateur du BDD, de concevoir, modéliser, générer et gérer visuellement des bases de données. Il comprend les outils pour créer des modèles ER (entité-relation) complexes et fournit également les fonctionnalités clés nécessaires aux tâches délicates de gestion et de documentation des changements qui exigent habituellement beaucoup de temps et d'efforts. [wiki\_02]



### 3.1. JOOMLA :

#### 3.1.1. Structures et Termes du Joomla<sup>2</sup> :

##### A. FRONTEND :

Par frontend nous entendons les zones du site Web que les visiteurs ou les utilisateurs enregistrés peuvent voir.

##### B. BACKEND :

C'est l'espace d'administration. Vous pouvez donner aux utilisateurs enregistrés le droit de travailler dans votre backend. Ce privilège est généralement limité à quelques employés, qui sont chargés d'administrer certaines tâches sur le site. Vous pouvez accéder à la page de connexion de votre administration via /administrator. Là, vous pouvez vous connecter avec vos identifiants de connexion et choisir votre langue préférée. (Figure 4).



Figure 4 : Joomla Administration registration

Une fois connecté, vous aurez accès à l'administration, qui est structurée en accord avec vos droits utilisateur (Figure 5).

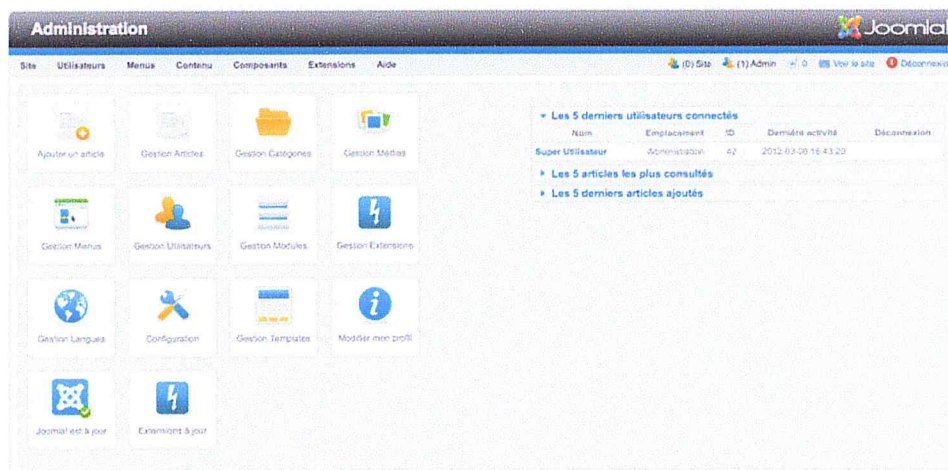


Figure 5: Administration

<sup>2</sup> paragraphe pris du : Star.R, 2010.

Vous pouvez accéder à chaque fonction, soit en utilisant le système de menus (et sous menus) soit en cliquant simplement sur une icône de l'écran de démarrage.

### C. FICHIERS :

Joomla! Se compose de centaines de fichiers: Images, scripts PHP, fichiers CSS, fichiers JavaScript, et beaucoup d'autres. Vous avez en fait déjà installé deux paquets Joomla!: un pour le frontend et un pour le backend. Le «Backend Joomla!» est situé dans le dossier /administrator .Ce dossier est recherché quand vous appelez /administrator dans votre navigateur. Dans ce dossier, il y a d'autres dossiers comme cache, composants, language, modules et templates. Les dossiers spécifiques au backend sont rangés dans ces répertoires.Vous trouverez les mêmes noms de dossier de nouveau en dehors du dossier administrator. Ces dossiers contiennent les fichiers du frontend. Ce ne sont pas réellement deux paquets Joomla!, mais il y a une séparation claire entre backend et frontend.

### D. LES ÉLÉMENTS DE JOOMLA:

Joomla! suppose que vous voulez écrire un article. Un article est constitué en général d'un titre, du texte et de quelques réglages de configuration.

- **Articles :**

Les articles peuvent être affichés seuls ou en liste. Sur la page d'accueil du site Joomla! que vous avez récemment installé vous devez voir quelque chose comme (*Figure 6*).

**Joomla!**

Félicitations, vous venez de créer un site Joomla!.  
Joomla rend facile la création d'un site tel que vous le rêvez et simplifie les mises à jour et la maintenance.  
Joomla est une plateforme flexible et puissante, que vous ayez besoin de créer un petit site pour vous-même ou un énorme site recevant des centaines de milliers de visiteurs.  
Joomla est Open Source, ce qui signifie que vous pouvez l'utiliser comme vous le souhaitez.

**Débutants**

Si vous vous lancez dans votre premier site Joomla, voir votre premier site web, vous êtes au bon endroit ! Joomla va vous aider à créer votre site web, d'une manière rapide et aisée.  
Commencez à utiliser votre site en vous connectant à l'administration avec l'identifiant et le mot de passe du compte que vous avez créé lors de l'installation de Joomla.

[Lire la suite : Débutants](#)

**Habités**

Si vous êtes un habitué de Joomla! 1.5, la version 2.5 vous paraîtra très familière. Elle possède de nouveaux templates et une interface utilisateur améliorée, mais la plupart des fonctionnalités sont identiques. Les changements les plus importants sont l'amélioration du contrôle d'accès (ACL) et celle de la gestion multi-niveaux des catégories.

[Lire la suite : Habités](#)

**Professionnels**

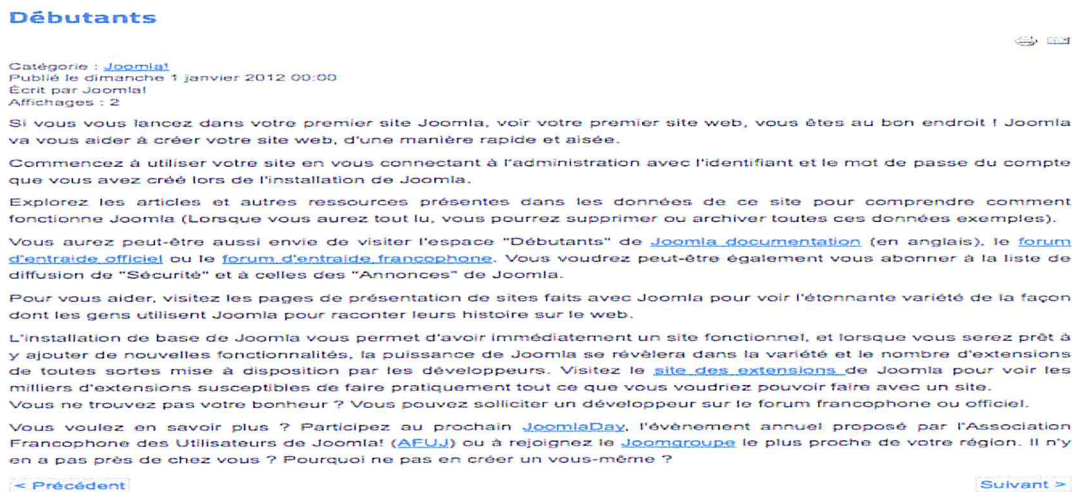
Joomla 2.5 est, dans la continuité du développement du Framework Joomla, un moyen puissant et flexible de transformer votre projet web en réalité. Avec son administration désormais totalement MVC, la possibilité de contrôler son aspect et la gestion de ses extensions est maintenant complète.

[Lire la suite : Professionnels](#)

Figure 6: Articles en page d'accueil

Les articles sont triés d'une certaine manière. Le premier article est affiché en utilisant toute la largeur du site. Les autres articles sont placés en dessous en trois colonnes. Si les articles sont trop longs, vous

pouvez insérer un lien lire la suite. Cette représentation est un affichage en liste. En cliquant sur le lien lire la suite vous serez redirigé sur une page affichant cet article seul (Figure 7). Le type d'affichage peut être modifié en définissant des options dans le backend.

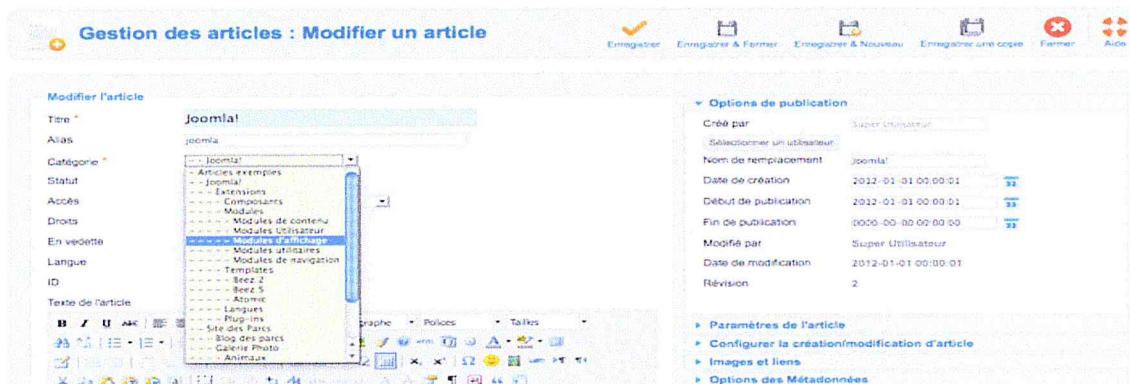


**Figure 7:** Affichage d'un article seul

Les articles peuvent être publiés ou non-publiés. Vous pouvez présenter des articles sur votre page d'accueil, vous pouvez les archiver ou les mettre à la poubelle et les récupérer. Vous pouvez les copier et les déplacer.

- **Catégories :**

Afin d'afficher clairement les articles, vous pouvez créer des catégories, puis leur affecter un article. Chaque article peut être assigné à exactement une catégorie (Figure 8). Les catégories peuvent être imbriquées jusqu'à n'importe quelle profondeur.



**Figure 8:** Affectation des catégories

- **Utilisateur :**

Les utilisateurs sont nécessaires pour produire du contenu. Il faut au moins un utilisateur enregistré sur chaque site Joomla, à savoir celui que vous avez créé lors de l'installation, avec les droits pour tout configurer sur votre site. En fonction des droits accordés, un utilisateur peut travailler dans le frontend et / ou le backend pour écrire un article. Chaque utilisateur a besoin d'un identifiant, une adresse email et d'un mot de passe. Chaque utilisateur peut être assigné à n'importe quel groupe d'utilisateurs ainsi qu'à n'importe quel niveau d'accès. Cela permet à l'utilisateur de créer des articles qui sont visibles seulement par certains groupes d'utilisateurs.

- **Navigation :**

Pour trouver votre chemin dans le site web, vous aurez besoin de navigation avec des liens correspondants. Dans Joomla! nous appelons cela un menu. Vous pouvez créer autant de menus que vous le désirez et les imbriquer en autant de façons différentes que vous le souhaitez. Chaque menu est un module qui peut être positionné sur une zone prévue dans le template.

- **Module :**

Un module est quelque chose que vous pouvez positionner à côté d'un article. Un menu, par exemple, est un module. Le petit bloc d'inscription sur le côté gauche est également un module. Vous pouvez créer autant de modules dotés de fonctions intelligentes selon vos besoins, et les placer sur des zones prédéfinies dans le Template.

- **Templates :**

Un template est une organisation graphique pour votre site web. La plupart des fichiers qui le composent sont des fichiers HTML et CSS. Joomla! offre plusieurs templates parmi lesquels vous pouvez faire votre choix. Les templates sont configurables, ce qui vous permet de charger un logo différent, changer la couleur de fond, etc. Chaque template offre des zones où les modules peuvent être positionnés.

- **Plug-ins :**

Un plug-in offre des services pratiques, mais est généralement invisible pour les visiteurs du site. Un éditeur WYSIWYG (What You See Is What You Get en anglais – Ce que vous voyez est ce que vous obtenez en français), par exemple, est un plug-in. Les Plug-ins sont des extensions, qui peuvent être installées un nombre de fois illimité. Le paquet du noyau comprend déjà de nombreux Plug-ins.

- **Composants :**

C'est une mini application intégrée à votre site Joomla!, disposant le plus souvent de sa propre interface de configuration dans la console d'administration Joomla!, et qui apporte de nouvelles fonctionnalités parfois très utiles.

### E. TYPES DE CONTENUS :

Dans le noyau Joomla! vous avez des *articles*, *catégories*, *liens web*, *bannières*, *contacts* et des *flux*. Souvent, vous voudrez enrichir votre contenu avec des fichiers comme des photos et d'autres types de médias. Pour cela Joomla! a son propre gestionnaire de médias.

- **Création d'un nouvel article :**

Lorsque vous vous connectez dans l'administration, vous voyez une icône appelée Ajouter un nouvel article. Vous pouvez soit cliquer sur cette icône soit accéder au formulaire via le menu principal *Contenu* ->*Gestion des articles* ->*Ajouter un nouvel article* (Figure 9).

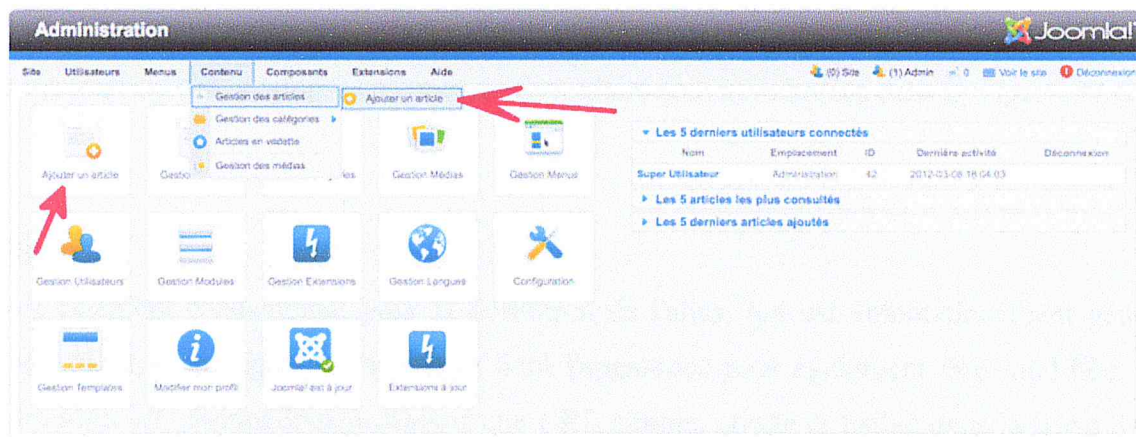


Figure 9: Administration

- **Formulaire d'article**

Vous êtes maintenant dans le formulaire d'article, par lequel vous allez écrire tous les articles que vous aurez à écrire dans Joomla!. Le formulaire est structuré comme dans la (Figure 10).

L'éditeur transforme votre texte écrit en format texte HTML. Joomla! utilise l'éditeur TinyMCE dans la configuration par défaut. TinyMCE est un projet open source indépendant, Il fonctionne essentiellement comme n'importe quel autre traitement de texte. Les boutons en dessous de la zone de saisie sont inhabituels et peuvent parfois prêter à confusion. Ces boutons sont spécifiques à Joomla! et ont seulement une relation indirecte avec l'éditeur. Le noyau de Joomla! offre cinq boutons supplémentaires. Vous pouvez ajouter d'autres boutons en ajoutant des extensions Joomla:

- a. **Article:** Permet de faire un lien vers d'autres articles existants de votre site.
- b. **Image:** Permet d'insérer une image ou une photo existante via le Gestionnaire de Médias ou de charger une nouvelle image.
- c. **Saut de page:** Permet d'insérer un saut de page dans votre article.
- d. **Lire la suite:** Permet de stipuler l'endroit où placer le lien Lire la Suite Lire la Suite.
- e. **Basculer l'éditeur:** Active ou non l'éditeur. Si l'éditeur est désactivé, vous verrez le code HTML de votre article.

### 1. Paramètres :

Ici vous pouvez par exemple, spécifier qui a écrit l'article (*Créé par*). L'utilisateur qui a créé l'article a généralement les autorisations et la responsabilité des modifications ultérieurement à l'article.

### 2. Plus de paramètres :

Dans cette section vous pouvez configurer de nombreuses options en les activant/désactivant. Vous pouvez modifier la disposition de l'article pour répondre aux besoins des lecteurs et des moteurs de recherche.

### 3. Droits sur l'article :

La sixième et dernière zone concerne les droits pour cet article.

## F. BASE DE DONNÉES

En plus des fichiers (graphiques, documents, dossiers système, etc.), Joomla! a également besoin d'une base de données. Durant la procédure d'installation, l'installateur web de Joomla! crée 61 tables dans la base de données que vous avez spécifiée. L'ensemble du contenu sera géré dans ces tables. Le schéma (*Figure 11*) obtenu après l'utilisation des outils de « reverse engineering » montre le schéma de la BDD de Joomla dont on va détailler dans le chapitre suivant.

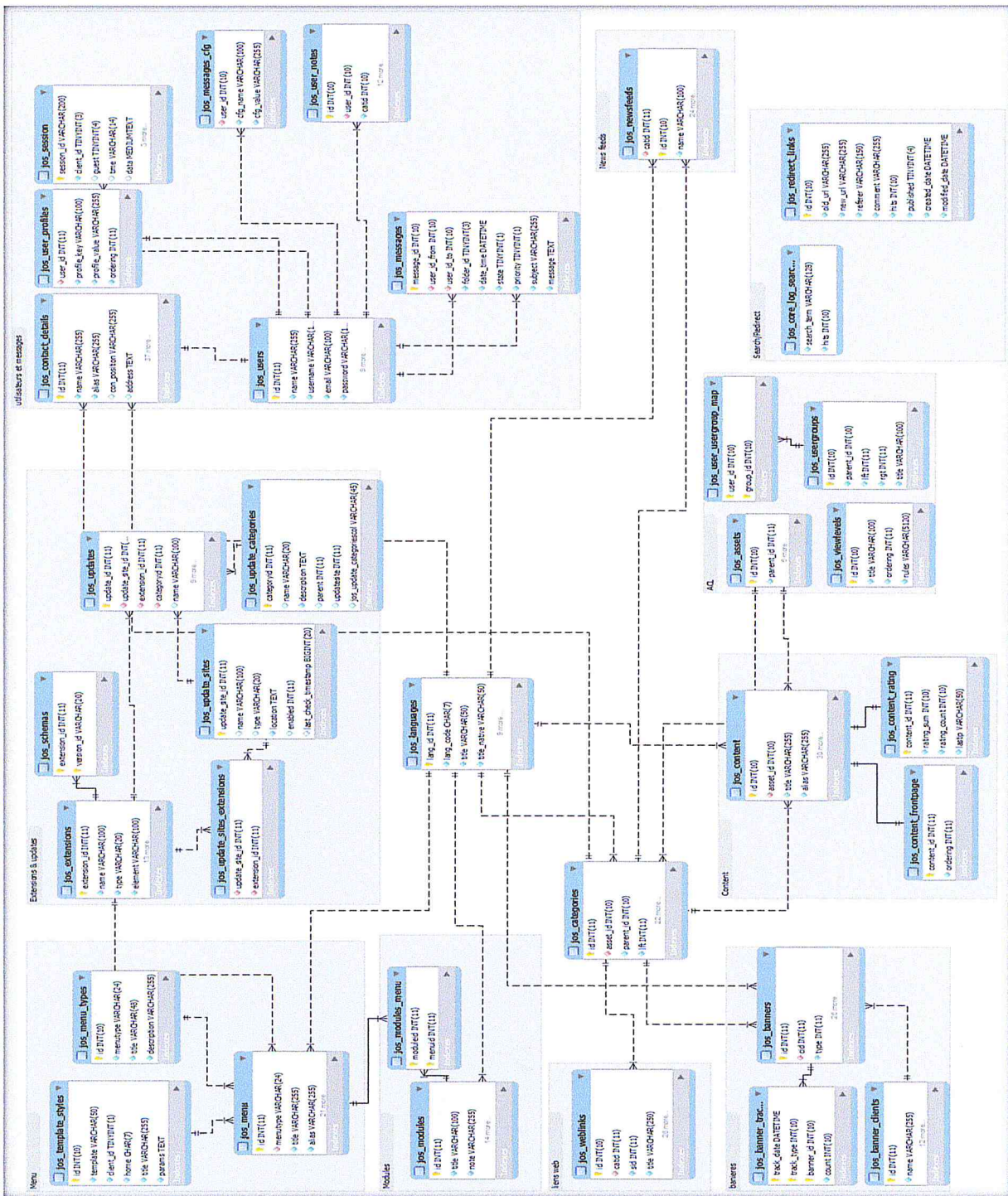


Figure 11 : modèle conceptuel de données « Joomla! 2.5 »

### 3.1.2. Architecture de Joomla

Joomla se repose sur 3 couches essentielles: [WEB\_07], [WEB\_08]

\_ **Couche d'extension** : qui regroupe les trois extensions modules, composants et templates et gestionnaire de langues.

\_ **Couche d'application** : présente l'ensemble des applications qui étend la classe application de Joomla et qui se charge chacune d'un rôle spécifique

- JInstallation qui se charge de l'installation de Joomla sur un serveur ;
- JAdministrator qui se charge de l'arrière-plan (back-end) d'administration ;
- JSite qui se charge du site (front-end) ;
- XML-RPC qui se charge de l'administration à distance du site Joomla.

\_ **Couche du Framework** : qui se compose de

- L'ensemble de classes qui composent le Framework de Joomla ;
- Les bibliothèques utilisées par le Framework ou ajoutées par les développeurs ;
- Les plugins qui étendent les fonctionnalités du Framework.

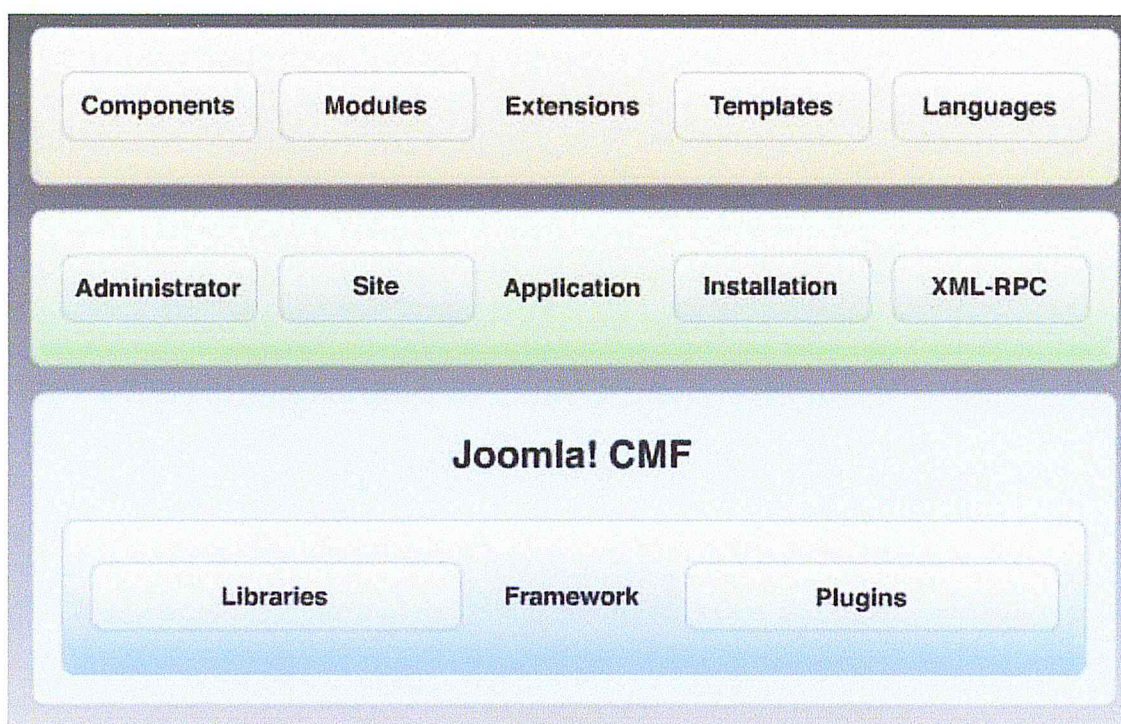


Figure 12: Joomla, architecture en couche.

L'image suivante présente avec plus de détails l'architecture de Joomla qui répond à l'architecture que doit avoir chaque CMS.



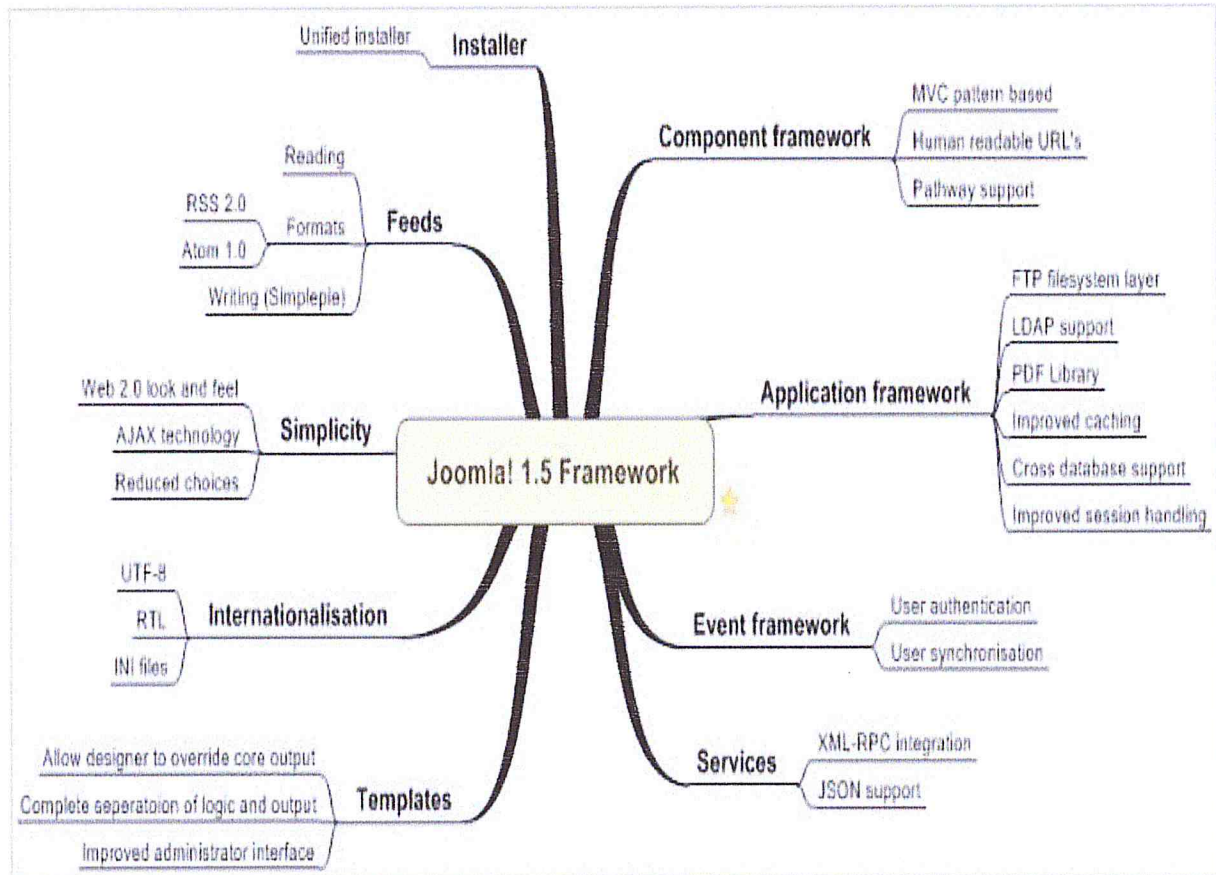


Figure 13: Joomla, architecture en arbre

### 3.1.3. Modèle MVC Joomla :

Bien que l'idée derrière un composant peut sembler très simple, le code peut devenir rapidement très complexe que des fonctionnalités supplémentaires sont ajoutées ou l'interface est personnalisée. Modèle-Vue-Contrôleur (MVC) est un modèle de conception de logiciel qui peut être utilisé pour organiser le code de telle manière que le logique métier et la présentation de données sont séparées. Le principe derrière cette approche est que si le logique métier est regroupé en un seul article, l'interface et l'interaction de l'utilisateur qui entoure les données peuvent être révisés et adaptés sans avoir à reprogrammer le logique métier. MVC a été initialement développé pour cartographier l'entrée traditionnelle, le traitement, les rôles de sortie dans une architecture logique GUI. Ces trois rôles principaux sont la base de la MVC Joomla. Ils sont décrits brièvement ci-après. [Dex\_01]

- **Modèle :**

Le modèle est la partie du composant qui encapsule les données de l'application. Il sera souvent fournir des routines pour gérer et manipuler ces données d'une manière significative en plus de routines qui récupèrent les données du modèle. Dans notre cas, le modèle contient des méthodes pour

ajouter, supprimer et mettre à jour des. En général, la technique d'accès aux données sous-jacente doit être encapsulée dans le modèle. De cette façon, si une application doit être déplacée d'un système qui utilise un fichier plat pour stocker ses informations à un système qui utilise une base de données, le modèle est le seul élément qui doit être changé, pas le point de vue ou le contrôleur.

- **Vue :**

La vue représente la partie de l'élément qui est utilisé pour restituer les données à partir du modèle d'une manière qui est appropriée pour l'interaction. Pour une application basée sur le Web, la vue serait généralement une page HTML qui est renvoyée à l'utilisateur. La vue récupère des données à partir du modèle (qui lui est transmise par le contrôleur) et alimente les données dans un modèle qui est peuplé et présenté à l'utilisateur. La vue ne provoque pas les données qui doivent être modifiés en aucune façon, il n'affiche que les données récupérées à partir du modèle.

- **Contrôleur :**

Le contrôleur est chargé de répondre aux actions des utilisateurs. Dans le cas d'une application web, une action de l'utilisateur est (en général) une demande de page. Le dispositif de commande détermine que la demande est faite par l'utilisateur et répond de manière appropriée en déclenchant le modèle pour manipuler les données de manière appropriée et en passant le modèle dans la vue. Le dispositif de commande n'affiche pas les données dans le modèle, il ne se déclenche que dans le modèle qui modifie les données, puis passer le modèle dans la vue qui affiche les données.

### 3.1.4. Connexion MVC :

La figure (14) montre les composants de base utilisés dans Joomla. [Jom\_01]

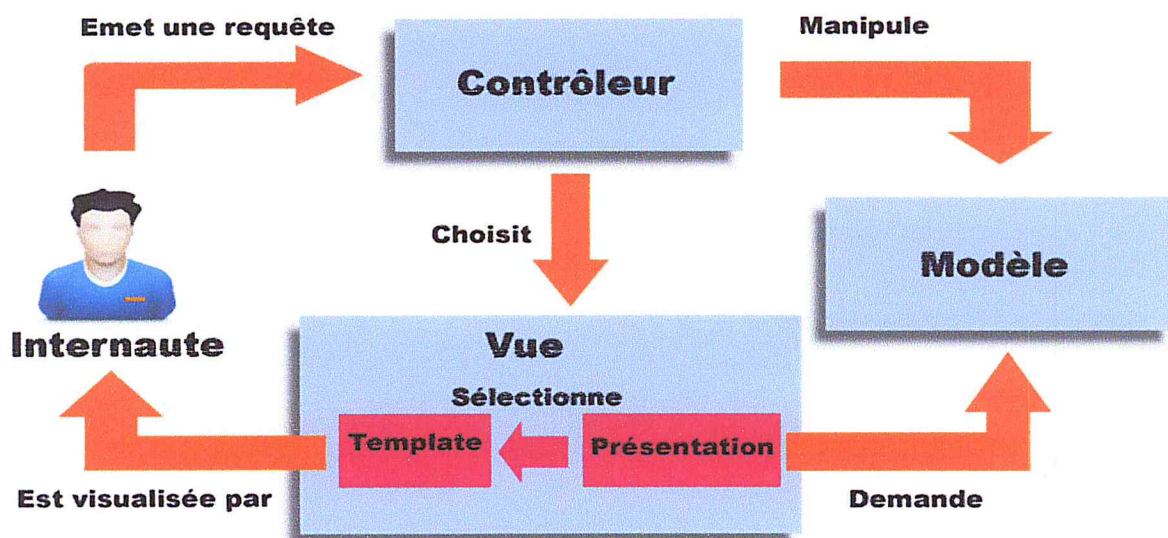


Figure 14 : Les modèles MVC de Joomla

### ✓ **Avantage :**

Un avantage apporté par ce modèle est la clarté de l'architecture qu'il impose. Cela simplifie la tâche du développeur qui tenterait d'effectuer une maintenance ou une amélioration sur le projet. En effet, la modification des traitements ne change en rien la vue. Par exemple on peut passer d'une base de données de type SQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne s'en trouvent pas affectées. [wiki\_03]

### **3.1.5. Conclusion :**

Le MVC est donc une des solutions vous permettant de maintenir aisément votre code, de le rendre plus clair et mieux organisé. Il n'est pas forcément la solution à tous les problèmes mais il constitue une base utilisée par de nombreux frameworks dans le monde.

## 3.2. WORDPRESS

### 3.2.1. Structure de WordPress:

#### A. FRONTEND :

Par frontend nous entendons les zones du site Web que les visiteurs ou les utilisateurs enregistrés peuvent voir.

#### B. TABLEAU DE BORD :

Le tableau de bord de WordPress est une sorte de page d'accueil pour l'administrateur du blog.

Pour arriver au tableau de bord WordPress, vous devez vous connecter (*figure 15*) à cette adresse [http://www.Votre\\_Site.fr/wp-login.php](http://www.Votre_Site.fr/wp-login.php).



Figure 15 : Connexion

-Une fois que vous êtes connecté vous arrivez sur le tableau de bord de WordPress:

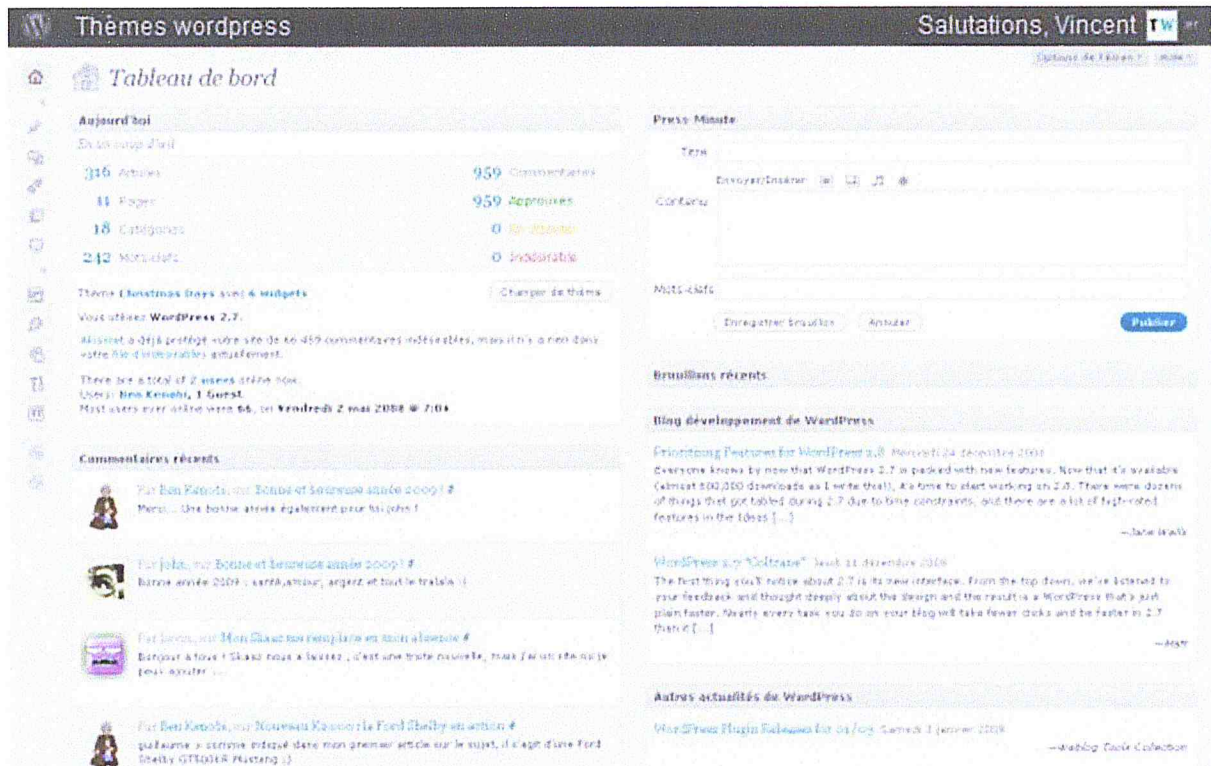


Figure 16 : Tableau de Bord (1)

Sur cette page s'affichent différentes informations du blog. Elle vous permet d'accéder à des fonctions importantes comme celle:

- D'accéder rapidement aux fonctions les plus courantes: Créer un nouvel article, modération des commentaires, Brouillon récent...
- Se tenir informé des dernières nouveautés WordPress.
- D'obtenir un aperçu rapide de votre blog: Nombre d'article, de page, de commentaire...

La colonne à gauche de l'écran fournit tous les liens pour accéder à la console d'administration de WordPress, avec les sous-menus qui s'affichent au survol:

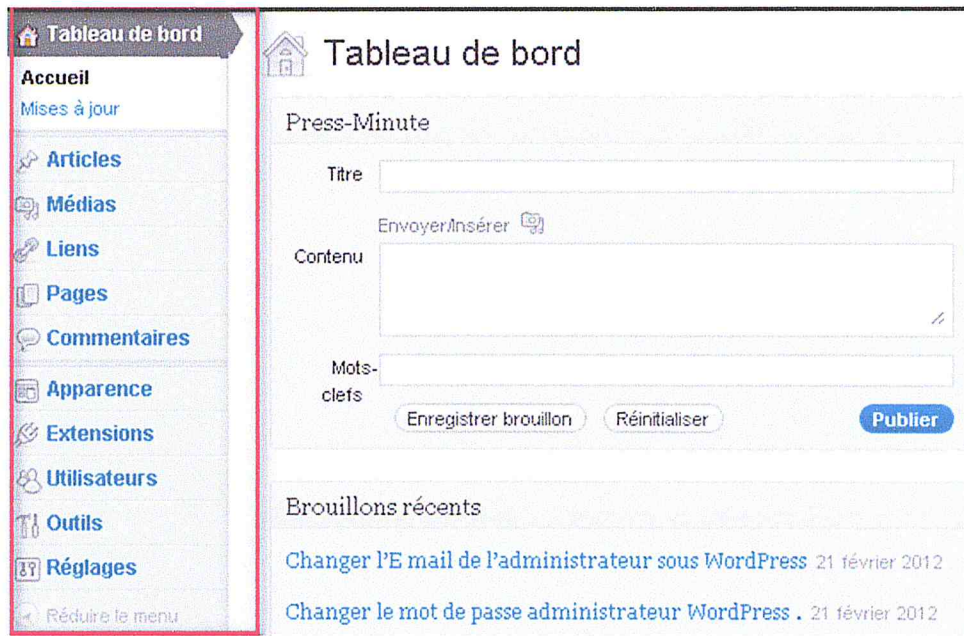


Figure 17 : Tableau de Bord (2)

### C. FICHIERS :

- Dossier d'application
  - fichiers de base (wp-settings.php, wp-config.php, etc.).
  - /wp-admin
    - Fonctionne comme un microsite pour contrôler l'application de sa propre css, js, comprend les répertoires.
  - /wp-includes
    - des classes, des bibliothèques, des scripts pour toute l'application, les bibliothèques JS, images.
  - /wp-content
    - /plug-ins
- PluginsPeuvent être soit des répertoires pour les grands plugins ou des fichiers simplement simples /themes.
  - Themes
- Annuaire des thèmes.
  - /uploads
- Généralement organisé par année / mois de téléchargement.

### D. BASE DE DONNÉES:

Le schéma (Figure 18) obtenu après l'utilisation des outils de « reverse engineering » montre le schéma de la BDD de wordpress dont il ya une courte description dessous.

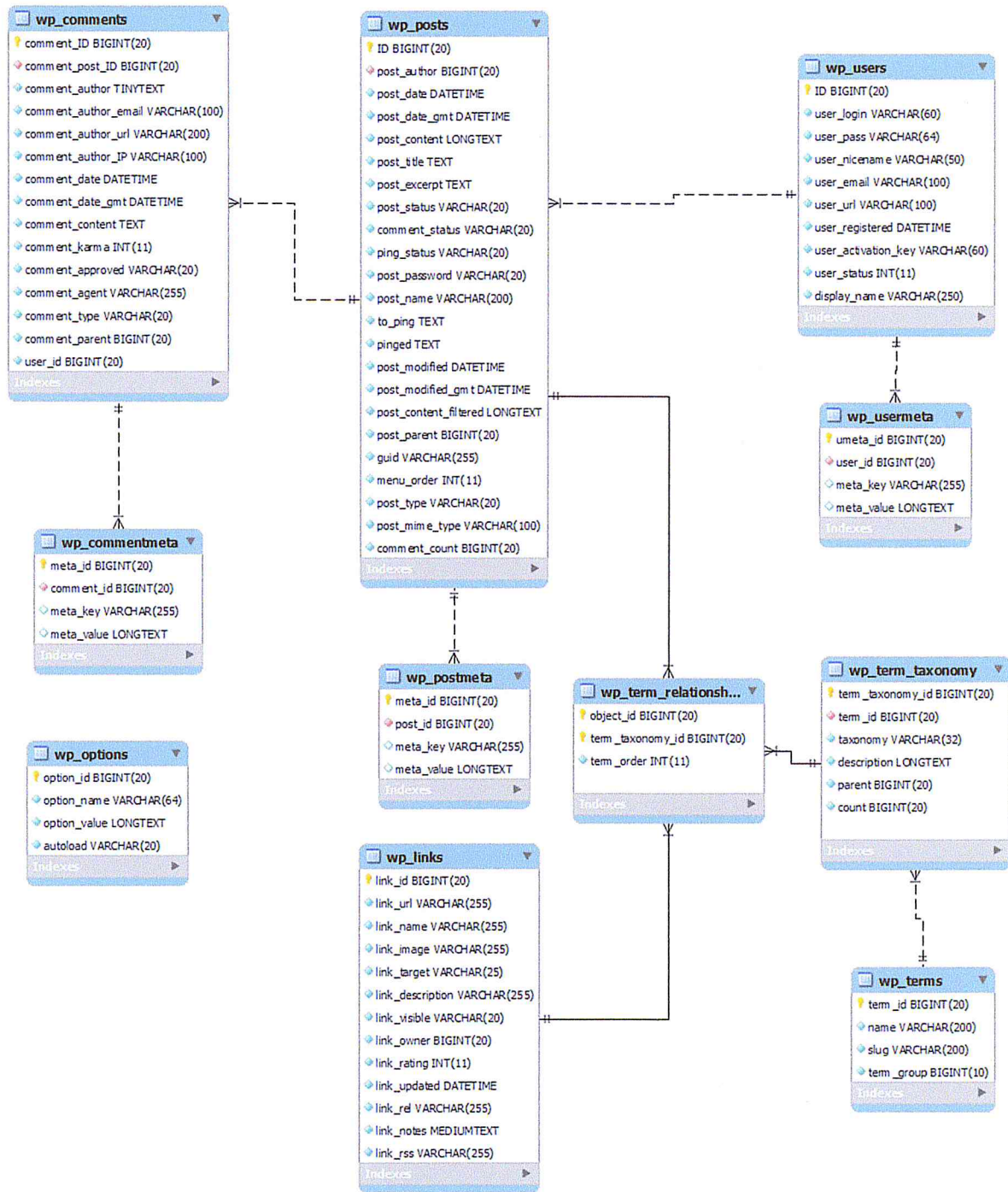


Figure 18 : modèle conceptuel de données « WordPress 3.5 »

### ➤ Description des tables:

wp\_comments

- wp\_commentmeta
- wp\_links
  - C'est la ou WP stocke les liens définis par l'utilisateur dans le panneau d'administration.
- wp\_options
  - C'est la ou WP stocke les qui figurent sous l'administration-> Paramètres.
- wp\_postmeta
  - Contient toutes les données supplémentaires (personnalisés) sur un poste / page.
- wp\_posts
  - les données nécessaires sur un poste / la page.
- wp\_terms
  - Définit les catégories, tags, catégories de liens et des taxonomies personnalisées. Dépend terms\_relationships et term\_taxonomy.
- wp\_term\_relationships
- wp\_term\_taxonomy
  - Définit le type de la taxonomie et contient des données sur la durée (nombre de poste, description, etc).
- wp\_usermeta
  - Contient toutes les données supplémentaires (personnalisés) sur un utilisateur.
- wp\_users
  - les données nécessaires sur un poste / la page.

### E. LES ELEMENTS DE WORDPRESS:

#### Thèmes WordPress :

Afin d'aider les utilisateurs à la construction de leur site avec des habillages originaux et facilement personnalisables sur le CMS Wordpress, il existe un large choix de thèmes Wordpress disponibles sur de nombreux sites.

#### 3.2.2. Architecture de WordPress :

Wordpress est écrit à l'aide de la programmation procédurale qu'est un paradigme de programmation basé sur le concept d'appel procédural. Une procédure, aussi appelée *routine* contient simplement une série d'étapes à réaliser. N'importe quelle procédure peut être appelée à n'importe quelle étape de



l'exécution du programme, y compris à l'intérieur d'autres procédures, voire dans la procédure elle-même (récursivité).

✓ **Avantage :**

La programmation procédurale est un meilleur choix qu'une simple programmation séquentielle, puisqu'elle permet la possibilité de réutiliser le même code à différents emplacements dans le programme sans avoir à le retaper (factorisation), ce qui a pour effet la réduction de la taille du code source et un gain en localité des modifications, donc une amélioration de la maintenabilité (compréhension plus rapide, réduction du risque de régression).

### 3.3. Conclusion Du Chapitre

La présente étude est donc pour but de comprendre comment chaque plateforme stocke ces données et quelle données vont être le sujet de migration et de trouver la méthodologie de la création d'un composant pour Joomla.

# CHAPITRE 4

Conception

Du

Traducteur

- **Introduction :**

En exploitant les résultats d'étude de chaque plateforme on est prêts à développer un composant MVC qui va migrer tous les données de Joomla vers WordPress.

### 4.1. Approches de migration :

**a. Migration via base de données :** l'utilisateur fournis les données pour connecter à la base de données du WordPress et exécute les fonctions d'insertion, mise à jours, transformations etc...Anis que les informations Pour accéder à un serveur ftp qui requière un nom d'utilisateur et un mot de passe, cette fonctionnalité peut être activé ou non du panneau de configuration du composant.

**b. Migration via un fichier XML :** cette méthode propose un énorme gain de temps, moins d'accès à la base de données et interruptible (indépendant de la connexion, une fois le fichier est téléchargé la migration est faite).Le fichier XML sera utilisé par le plugin du WordPress d'importation des données externes.

#### 4.1.1. Approche de migration des données via base de données :

Cette approche nécessite la construction des mcd du Joomla et du WordPress ainsi que les formats des données stocké dans chaque BDD.

Afin d'assurer le bon déroulement du processus de migration vers la base de données du WordPress, la base de données de ce dernier devrait être vide pour qu'il y aura pas d'erreur lors de l'exportation.

Le but est de concevoir et développer un composant qui prend en charge la migration de toutes les catégories, les messages, les images, les médias et les étiquettes (tags) de Joomla vers WordPress.

**Principales caractéristiques du composant:**

- \* Faire la migration de toutes les catégories.
- \* Upload tous les médias des articles dans le répertoire d'upload du WordPress.
- \* Modifie le contenu des articles pour garder les liens vers les médias.
- \* Maintient les attributs des images et autres media
- \* Modifie les liens internes pour pointez aux éléments du WordPress
- \* Migre les mots clés comme des tags (étiquettes)
- \*Migre auteurs et les autres utilisateurs.

Pour chaque types d'éléments qu'il va être migré, il faut comprendre comment il est stocké et gérer dans chaque plateforme, voici un exemple d'exportation d'articles:

➤ Migration des articles :

- Les articles sous Joomla contiennent un texte qui peut inclure des liens internes, externes et/ou des medias. [Paro\_01]

Figure 19 montre les tables du BDD du Joomla ayant des données relatives aux articles

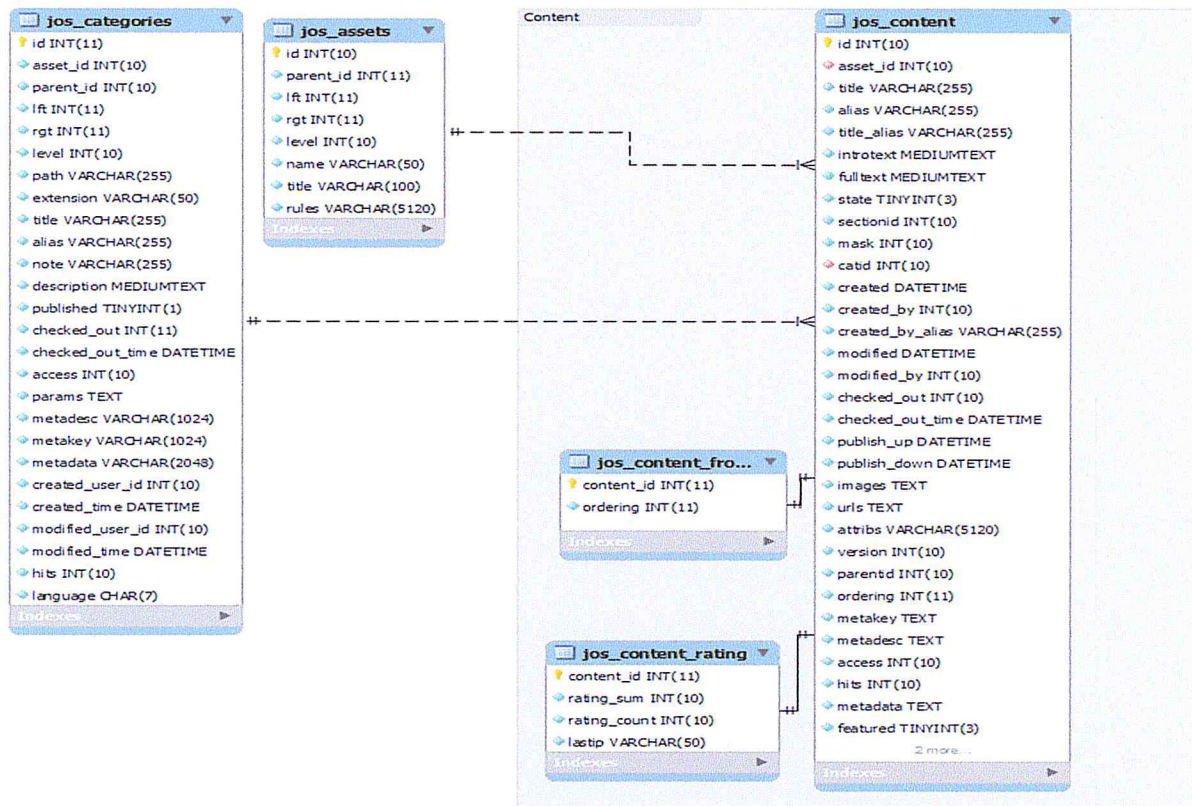


Figure 19 : Diagramme de classes partial « articles »

• Description des tables :

**jos\_assets** : Comprend une ligne pour chaque composant et pour chaque article qui dispose des autorisations d'ACL. La colonne `rules` –règles en français- stocke les permissions au format JSON. Ce tableau est utilisé chaque fois qu'on vérifie si un utilisateur est autorisé à effectuer une action.

**jos\_content\_frontpage** : Contient une ligne pour chaque objet affecté afficher en vedette du site.

**jos\_content\_rating** : Contient une ligne pour chaque article notées dans le site.

**jos\_content** : Contient une ligne pour chaque article défini dans le site.

**jos\_categories** : Contient une ligne pour chaque catégorie définie dans le site. Cela inclut article, bannière, contact, newsfeed, et les catégories des liens Internet.

La figure 20 montre les tables du BDD du wordpress ayant des données relatives aux 'posts'.

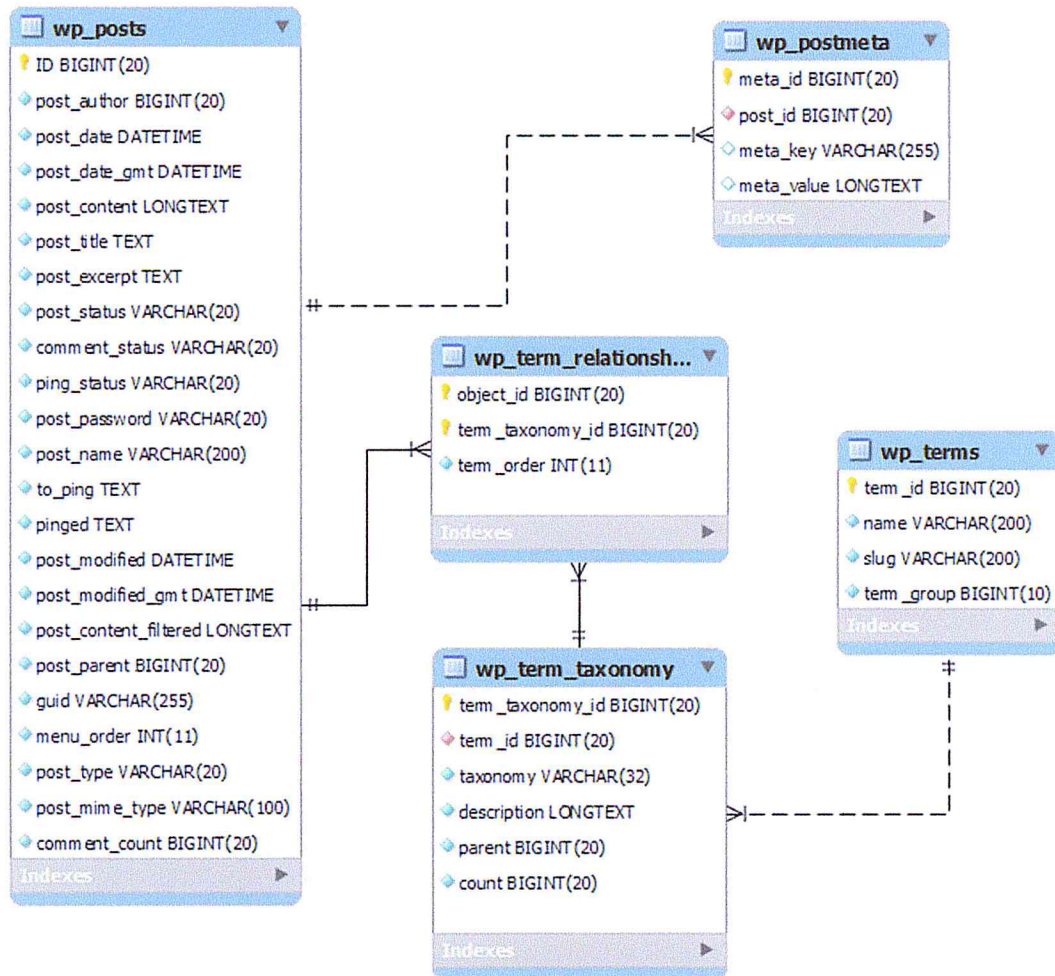


Figure 20 : Diagramme de classes partial 'posts'

• Description des tables :

**Wp\_posts** : Les données essentielles sur les articles, pages, les éléments de menu de navigation ainsi que les informations sur les attachements (medias).

**Wp\_postmeta** : Chaque article a des informations et des caractéristiques appelé les méta-données et ils sont stocké dans le wp\_postmeta. Certains plugins peuvent ajouter leurs propres informations à ce tableau.

**Wp\_terms** : Les catégories pour les articles, les liens et les étiquettes (tags) pour les postes se trouvent dans le tableau wp\_terms.

**Wp\_tersms\_relationships** : les articles sont associés à des catégories et les étiquettes de la table wp\_terms et cette association est maintenue dans le tableau des wp\_term\_relationships. L'association de liens vers leurs catégories respectives et sont également conservés dans ce tableau.

**Wp\_term\_taxonomy** : Ce tableau décrit la taxonomie (catégorie, un lien ou tag) pour les entrées de la table wp\_terms.

• **Mappage des données :**

Pour faire migrer un article du Joomla à wordpress ainsi que toutes ces données relatives, il faut faire la correspondance entre chaque attribut. Figure 21 montre un exemple de la correspondance entre la table « jos\_content » et les tables qui vont contenir les données de ce dernier.

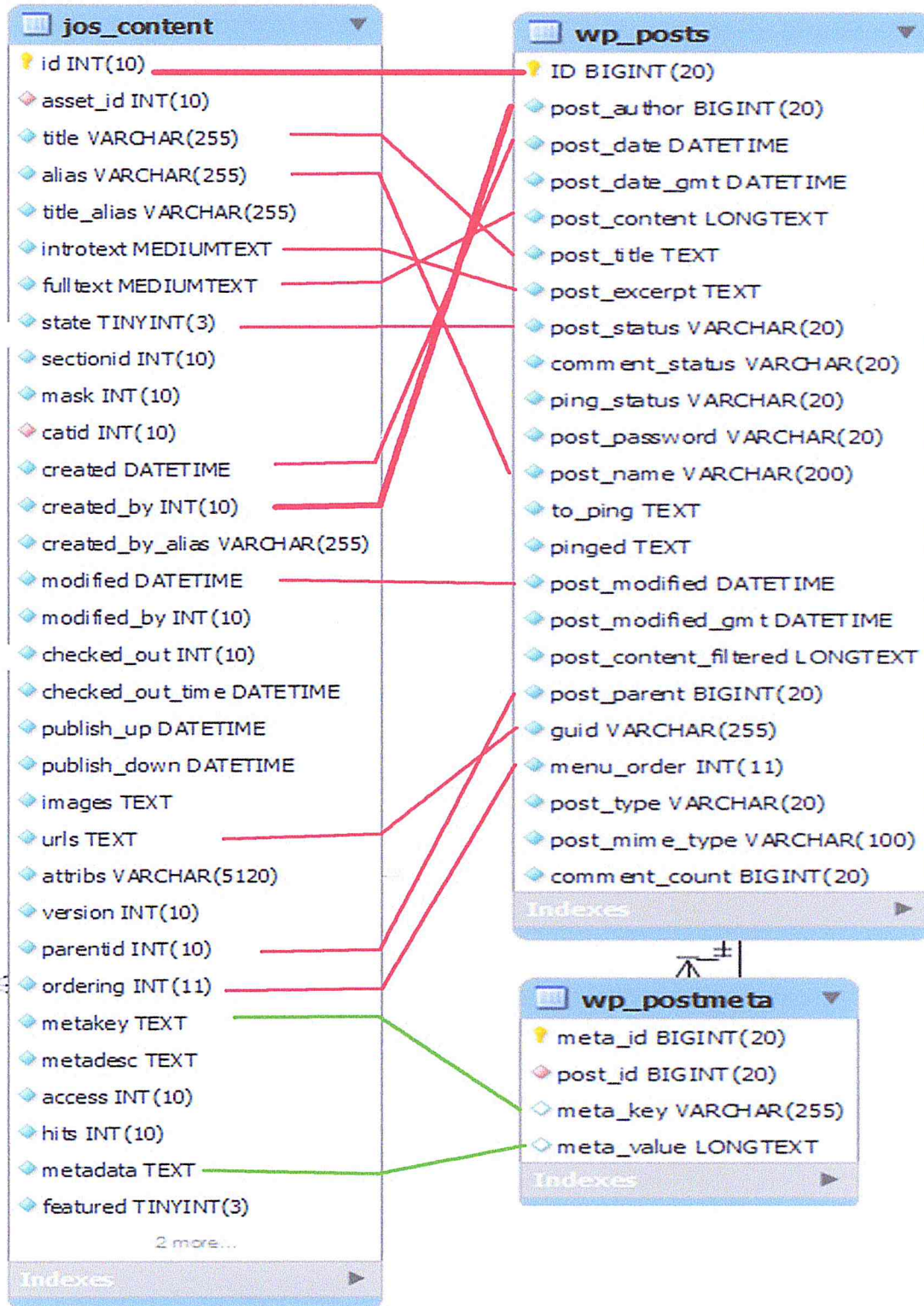


Figure 21 : exemple de mappage des données

### • Traiter le contenu de l'article :

L'article sous Joomla ! contient des liens internes vers d'autres éléments de Joomla ! Et des liens d'insertions des medias ce qui implique qu'avant de migrer l'article vers la nouvelle BDD il faut modifier les liens existants, pour cela il faut repérer les liens et les modifier ce qui implique l'utilisation des expressions régulières.

- **Expression régulière (ReGeX):** est une chaîne de caractères que l'on appelle parfois un motif et qui décrit un ensemble de chaînes de caractères possibles selon une syntaxe précise. Les ReGeX servent à faire correspondre des chaînes de caractères sur un flux de données. [Lar\_01]

### Habitués



Catégorie : [Joomla!](#)

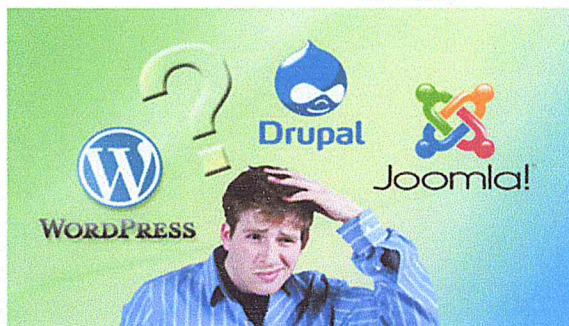
Publié le dimanche 1 janvier 2012 00:00

Écrit par Super Utilisateur

Affichages : 6

Si vous êtes un habitué de Joomla! 1.5, la version 2.5 vous paraîtra très familière. Elle possède de nouveaux templates et une interface utilisateur améliorée, mais la plupart des fonctionnalités sont identiques. Les changements les plus importants sont l'amélioration du contrôle d'accès (ACL) et celle de la gestion multi-niveaux des catégories.

Vous pourrez en savoir plus en lisant [l'article](#).



Le nouveau gestionnaire d'utilisateurs vous permettra de gérer qui a accès à quoi dans votre site. Vous pouvez conserver l'organisation des groupes d'accès tel que vous la connaissez dans Joomla! 1.5 ou la rendre aussi complexe que nécessaire. Vous pourrez en savoir plus sur [comment fonctionne le contrôle d'accès](#) sur le [site de documentation Joomla!](#) (liens en anglais).

Pour en savoir plus sur comment migrer un site Joomla! 1.5 en version 2.5, lisez cet [article](#).

[< Précédent](#)

[Suivant >](#)

Figure 22 : article à exporter

Avant d'exporter cette article il faut transformer les liens internes qui pointent vers d'autres éléments de Joomla ! (Liens en marron figure 22) pour fonctionner dans la nouvelle plateforme WordPress, et capturer les médias et les transférer dans le répertoire des média de WordPress. Les liens en jaune sont des liens externes qui ne vont pas être transformé.

### 4.1.2. Approche de migration des données via fichier XML :

Parfois une base de données wordpress vide est simplement impossible, Par exemple : un site wordpress déjà existant et que l'administrateur veut importer des données de joomla ! Pour enrichir son site ou pour de s'assurer du « backup » des données du site : textes, images, liens... etc et/ou de la portabilité éventuelle des données vers un autre espace d'hébergement, wordpress offre nativement des possibilités « d'import » et « d'export » au format XML, ou la connexion direct entre les deux bases de données est impossible, pour cela la génération d'un fichier XML devient utile.

Après avoir étudié les scripts php des plugins « export » et « import » du wordpress on a parvenu à comprendre la structure du fichier ainsi que la structure de chaque éléments a exporté.

#### ➤ Structure du fichier :

- Après avoir étudié la structure de fichier, il ne reste que faire les transformations des données et faire les requêtes SQL pour générer le fichier d'exportation XML le text suivant est un exemple (partie de fichier xml d'exportation).

```

• <?xml version="1.0" encoding="UTF-8"?>
• <rss version="2.0"
•     xmlns:excerpt="http://wordpress.org/export/1.0/excerpt/"
•     xmlns:content="http://purl.org/rss/1.0/modules/content/"
•     xmlns:wfw="http://wellformedweb.org/CommentAPI/"
•     xmlns:dc="http://purl.org/dc/elements/1.1/"
•     xmlns:wp="http://wordpress.org/export/1.0/"
• >
• <channel>
• <!-- information sur le site (blog) -->
• <channel>
•   <title/>
•   <link/>
•   <description/>
•   <pubDate/>
•   <language/>
•   <wp:wxr_version/>
•   <wp:base_site_url/>
•   <wp:base_blog_url/>
• <!-- pour chaque auteur -->
• <wp:author>
• <wp:author_id/>
• <wp:author_login/>
• <wp:author_email/>

```



- `<wp:author_display_name></wp:author_display_name>`
- `<wp:author_first_name></wp:author_first_name>`
- `<wp:author_last_name></wp:author_last_name>`
- `</wp:author>`
- `<!-- pour chaque categorie -->`
- `<wp:category><wp:term_id></wp:term_id>`
- `<wp:category_nicename></wp:category_nicename>`
- `<wp:category_parent></wp:category_parent>`
- `</wp:category_name></wp:category_name>`
- `</wp:category_description></wp:category_description>`
- `</wp:category>`
- `<!-- pour chaque étiquette (tag) -->`
- `<wp:tag><wp:term_id></wp:term_id>`
- `<wp:tag_slug></wp:tag_slug>`
- `</wp:tag_description></wp:tag_description>`
- `</wp:tag>`
- `<!-- pour chaque terme -->`
- `<wp:term><wp:term_id></wp:term_id>`
- `<wp:term_taxonomy></wp:term_taxonomy>`
- `<wp:term_slug></wp:term_slug>`
- `<wp:term_parent></wp:term_parent>`
- `</wp:term_name></wp:term_name>`
- `</wp:term_description></wp:term_description>`
- `</wp:term>`
- `<!-- pour chaque tuple de la table wp_posts -->`
- `<item>`
- `<title></title>`
- `<link><?php the_permalink_rss() ?></link>`
- `<pubDate></pubDate>`
- `<dc:creator></dc:creator>`
- `<guid isPermaLink=""> </guid>`
- `<description></description>`
- `<content:encoded></content:encoded>`
- `<excerpt:encoded></excerpt:encoded>`
- `<wp:post_id></wp:post_id>`
- `<wp:post_date></wp:post_date>`
- `<wp:post_date_gmt></wp:post_date_gmt>`
- `<wp:comment_status></wp:comment_status>`
- `<wp:ping_status></wp:ping_status>`
- `<wp:post_name></wp:post_name>`
- `<wp:status></wp:status>`
- `<wp:post_parent></wp:post_parent>`
- `<wp:menu_order></wp:menu_order>`

- `<wp:post_type></wp:post_type>`
- `<wp:post_password></wp:post_password>`
- `<wp:is_sticky></wp:is_sticky>`
- `<!-- si le type du post est attachment-->`
- `<wp:attachment_url></wp:attachment_url>`
- `<!-- pour chaque term -->`
- `<category domain=\" \" nicename=\" \"></category>`
- `<!--metadata d'article -->`
- `<wp:postmeta>`
- `<!-- pour chaque clé-->`
- `<wp:meta_key></wp:meta_key>`
- `<wp:meta_value></wp:meta_value>`
- `</wp:postmeta>`
- `</item>`
- `</channel>`
- `</rss>`

• **Introduction :**

L'objectif de ce chapitre est de montrer comment installer, mettre en place et de tester le traducteur des deux approches.

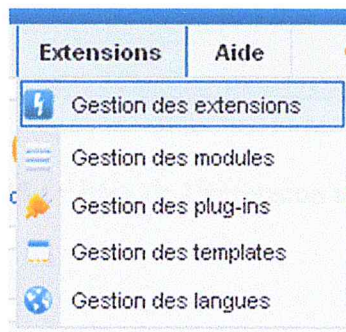


**5.1. Installation et mise en œuvre de la migration via BDD :**

**5.1.1. Installation Du Composant :**

L'installateur des composants Joomla! Est fourni nativement avec son lot d'extensions qui permettent de l'utiliser dès son installation, pour installer le composant suivez les étapes suivantes :

En cliquant dans le menu **Extensions - Gestion des extensions**



**Figure 23:** Gestion des extensions

➤ On arrive sur la page de l'installateur.

**Archive à transférer**

Archive

---

**Installer depuis un dossier**

dossier d'installation

---

**Installer depuis une adresse URL**

URL

**Figure 24 :** Installateur d'extensions du Joomla !

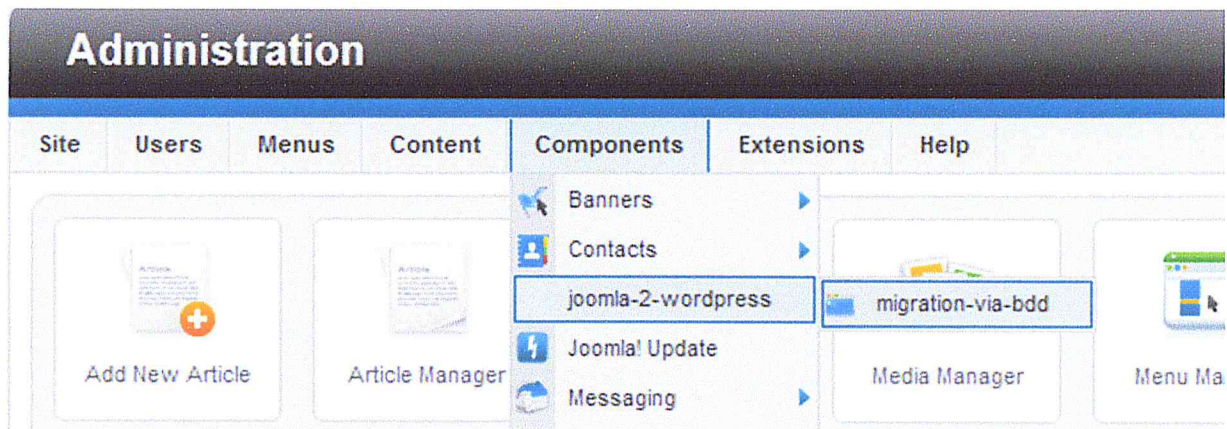


Figure 27: Accès au composant

➤ Une fois accédé au composant, un message d'accueil est affiché et vous demande de remplir un formulaire pour pouvoir connecter à la base de données du wordpress. Figure(28)

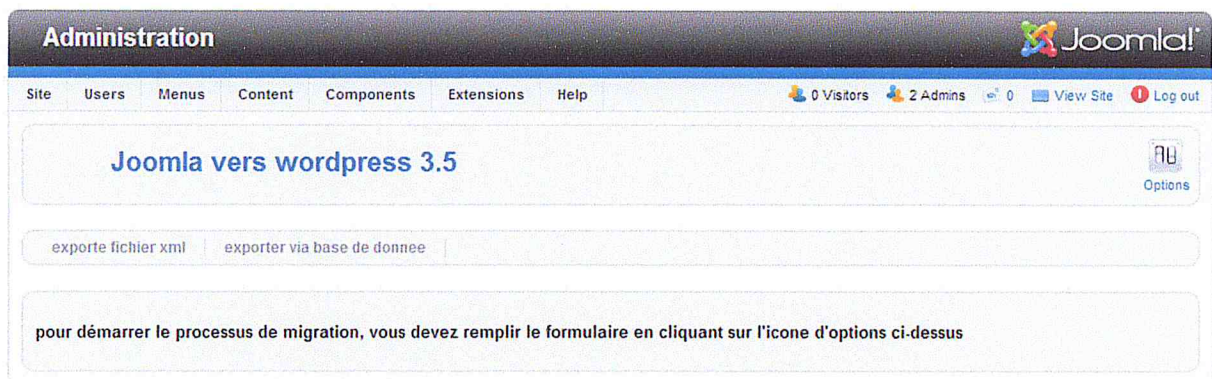


Figure 28: Ecran d'accueil du composant

-En cliquant sur l'icône d'options une fenêtre sera affichée. Figure (29).

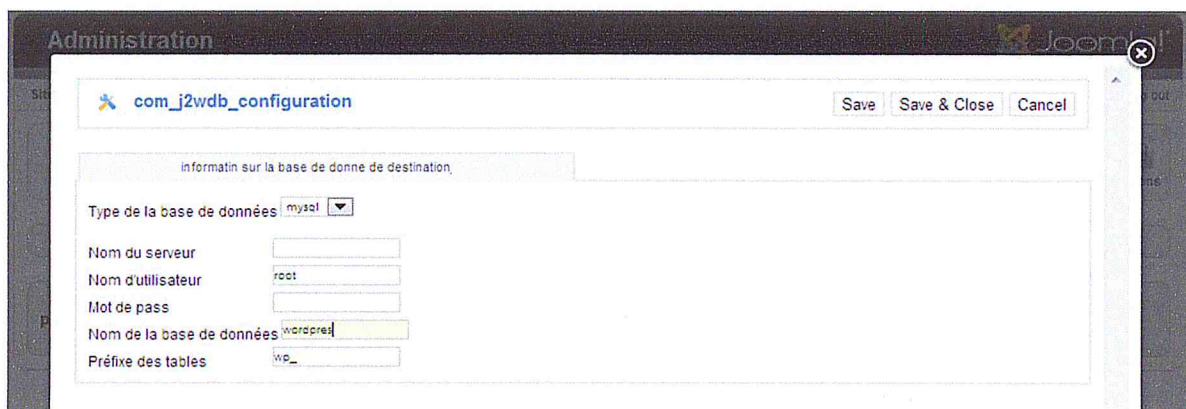


Figure 29: Informations sur la BDD de destination

➤ Une fois le formulaire remplis cliquer sure « save&close » si les données sont correct le composant connecte au bdd, migre les données qui seront affichés. Figure (30).

Utilisateurs export&						
<input type="checkbox"/>	nom	nom utilisateur	email	derniere visite	date de registration	id
<input type="checkbox"/>	Super User	admin	b_rafik@live.fr	2013-06-10 01:24:08	2013-06-02 08:55:52	674

Catégorie exporté				
<input type="checkbox"/>	title	access	Language	ID
<input type="checkbox"/>	Uncategorised	public	all	2
<input type="checkbox"/>	Blog	public	all	9

Articles exporté				
<input type="checkbox"/>	title	categorie	auteur	ID
<input type="checkbox"/>	About	Uncategorised	Super User	1
<input type="checkbox"/>	Working on Your Site	Uncategorised	Super User	2
<input type="checkbox"/>	Welcome to your blog	Blog	Super User	3
<input type="checkbox"/>	About your home page	Blog	Super User	4
<input type="checkbox"/>	Your Modules	Blog	Super User	5

Figure 30 : Les données migrées

### 5.1.3. Test :

- **Affichage :**

La Figure (31) montre le gestionnaire des articles sous Joomla ! (trié par ID) :

<input type="checkbox"/>	Title	Status	Featured	Category	Ordering	Access	Created by	Date	Hits	Language	ID
<input type="checkbox"/>	About (Alias: about)	✓	+	Uncategorised	1	Public	Super User	2012-01-04	4	All	1
<input type="checkbox"/>	About your home page (Alias: about-your-home-page)	✓	+	Blog	1	Public	Super User	2012-01-04	7	All	4
<input type="checkbox"/>	Welcome to your blog (Alias: welcome-to-your-blog)	✓	+	Blog	2	Public	Super User	2012-01-04	3	All	3
<input type="checkbox"/>	Working on Your Site (Alias: working-on-your-site)	✓	+	Uncategorised	0	Special	Super User	2012-01-04	7	All	2
<input type="checkbox"/>	Your Modules (Alias: your-modules)	✓	+	Blog	0	Public	Super User	2012-01-05	4	All	5

Figure 31 : Gestionnaire d'articles Joomla

La Figure (32) montre le gestionnaire des articles exporté sous wordpress (trié par date de la publication) :

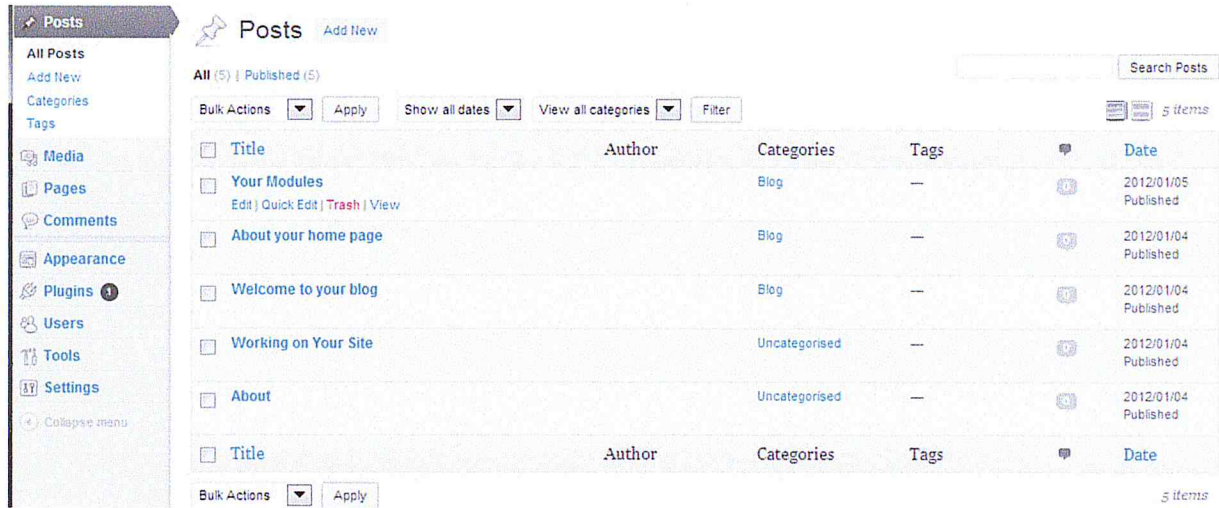


Figure 32 : Gestionnaire d'articles WordPress

- Exemple d'affichage d'un article sous Joomla !

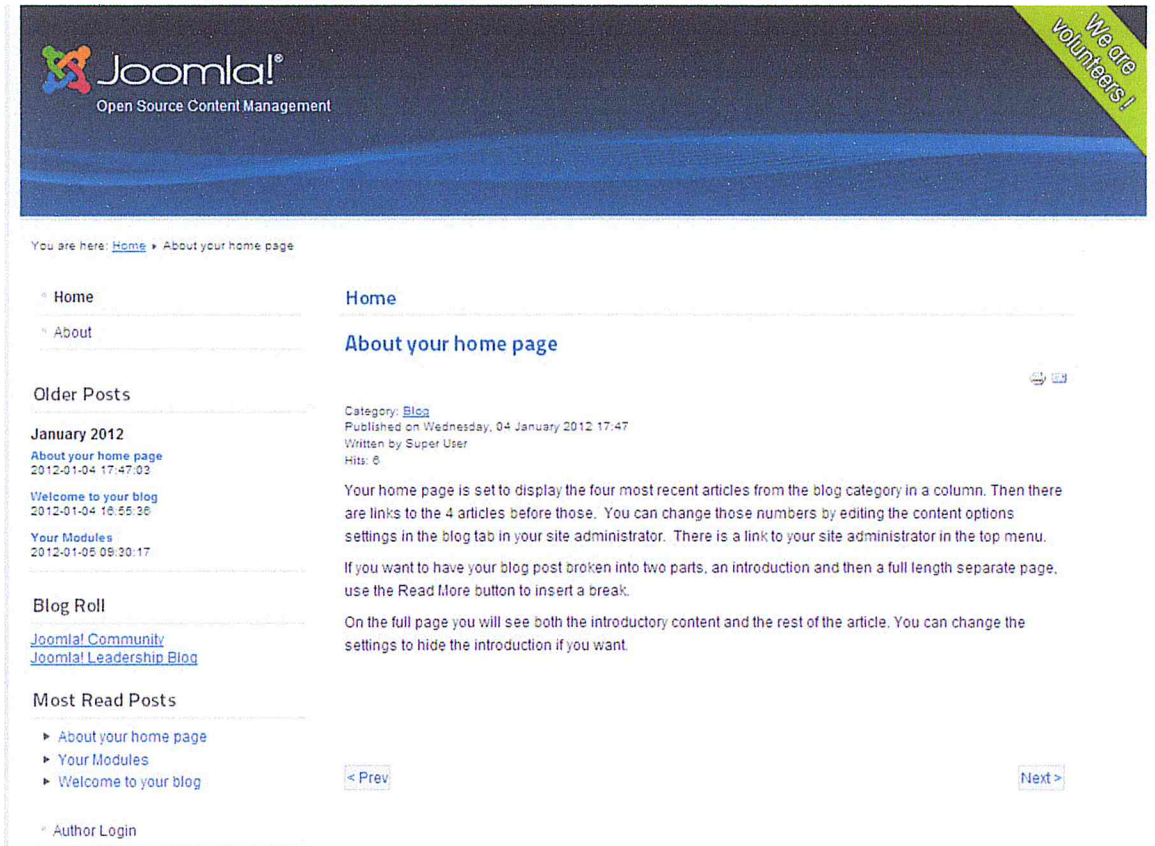


Figure 33 : Article affiché par Joomla !

Figure (34) montre l’affichage du même article par wordpress après le processus de la migration.

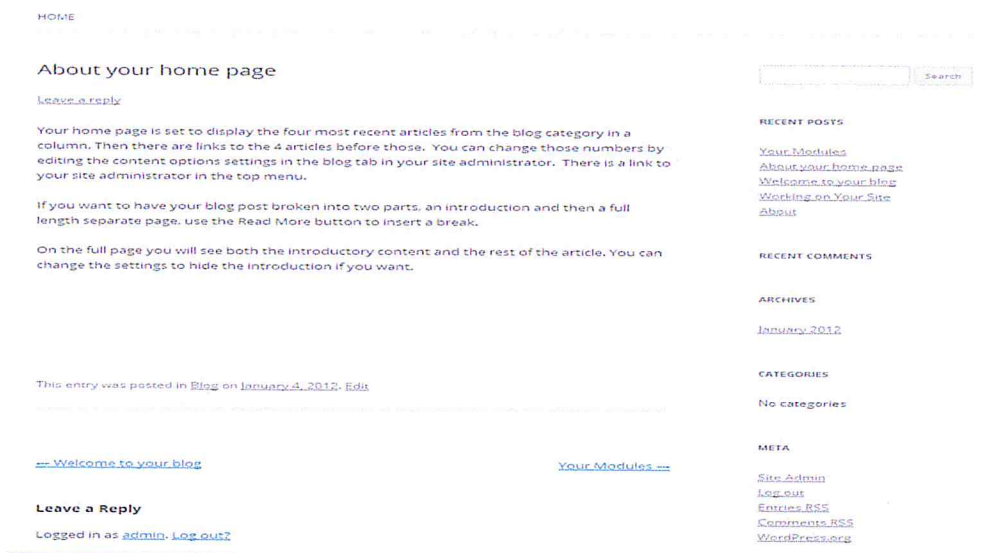


Figure 34 : Article migré de Joomla vers WordPress

## 5.2. Installation et mise en œuvre de migration via fichier XML :

### 5.2.1. Installation Du Composant :

Les mêmes étapes d’installation que le composant de migration via BDD (expliquer précédemment).

### 5.2.2. Les étapes de l’exportation du fichier XML :

Les étapes à suivre pour un export des données sous Joomla.

- Accédé au composant.
- Une fois accédé à l’interface du composant, ce dernier génère le fichier et il affiche un lien pour télécharger ce dernier. figure (35).

Vous possédez désormais un fichier XML d’archive contenant la totalité des données de votre Joomla !, à l’exception des fichiers média dont le plugin import du wordpress va les télécharger.

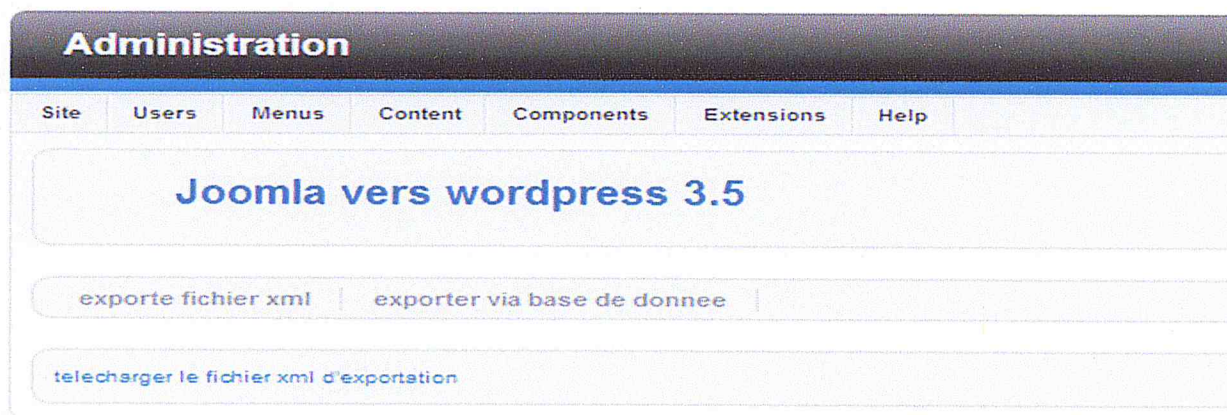


Figure 35 : Génération du fichier XML

### 5.2.3. Les étapes de l'importation du fichier XML :

Les étapes à suivre pour importer un fichier XML dans un WordPress.

- Il faut accéder dans le menu outils (figure 36).
- On choisit importer (figure 37).
- On choisit l'importation de contenu du WordPress, une nouvelle page apparaîtra, on clique sur parcourir pour choisir le fichier XML à importer, ensuite on clique sur OK puis envoyer le fichier et l'importer.
- On peut choisir ici de créer des nouveaux utilisateurs ou un utilisateur existant comme auteur des articles importé. Ensuite, on choisit de télécharger ou non les fichiers joints au contenu. Il ne reste qu'à cliquer sur envoyé.
- Une nouvelle page vous indique que l'importation a correctement été effectuée.

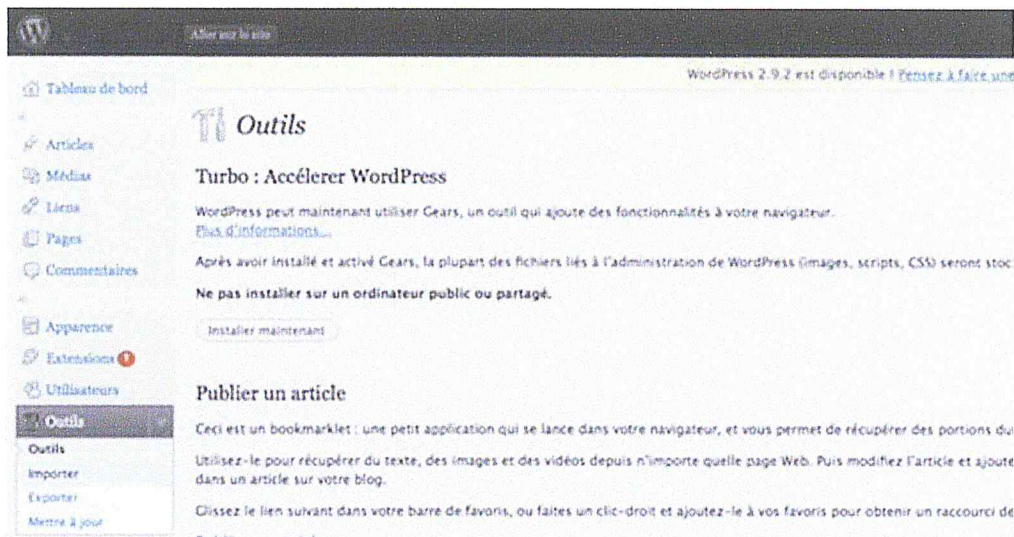


Figure 36 : Accéder au menu outils

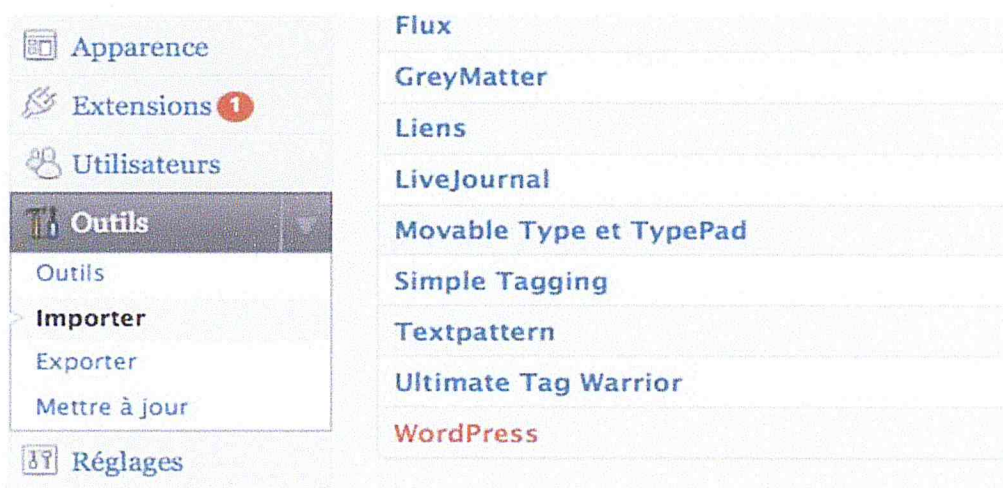


Figure 37 : Sélection du plugin « importer »



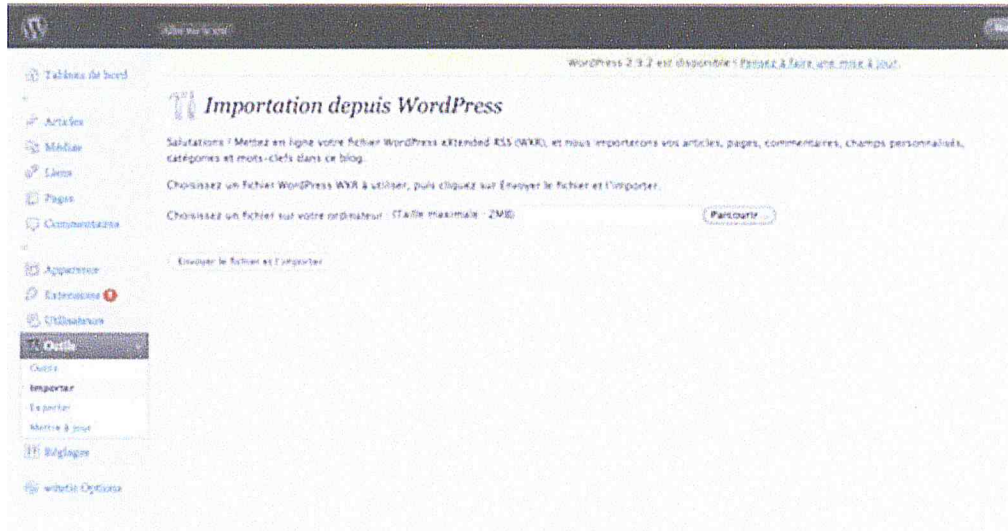


Figure 38 : Sélection du fichier d'exportation

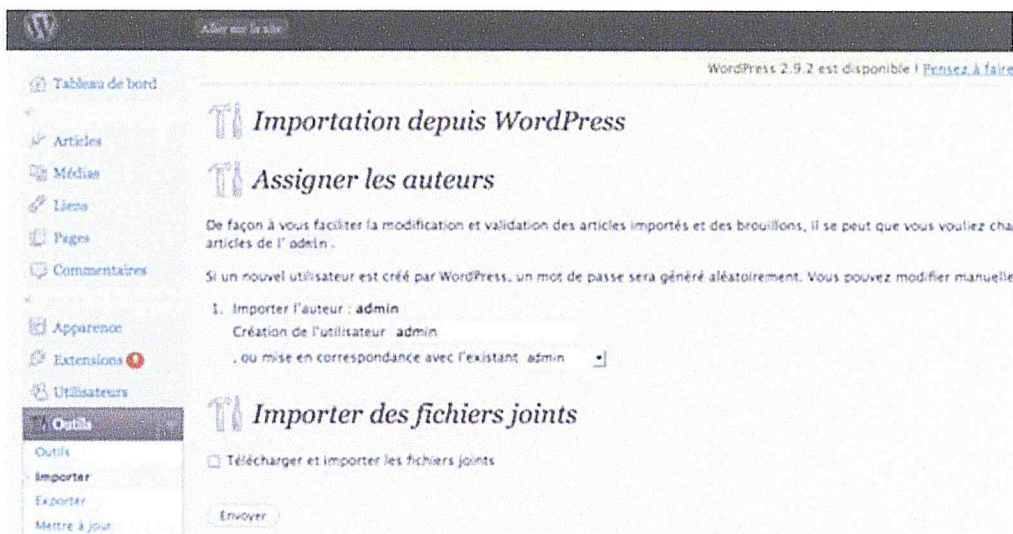


Figure 39 : Sélection des options d'exportation

#### 5.2.4. Test :

Pour le test on a créé un article sous Joomla (figure 40), ensuite on a téléchargé le fichier XML et ce dernier est importé via le plugin d'importation du wordpress ce qu'est a donné le résultat du (figure 41).

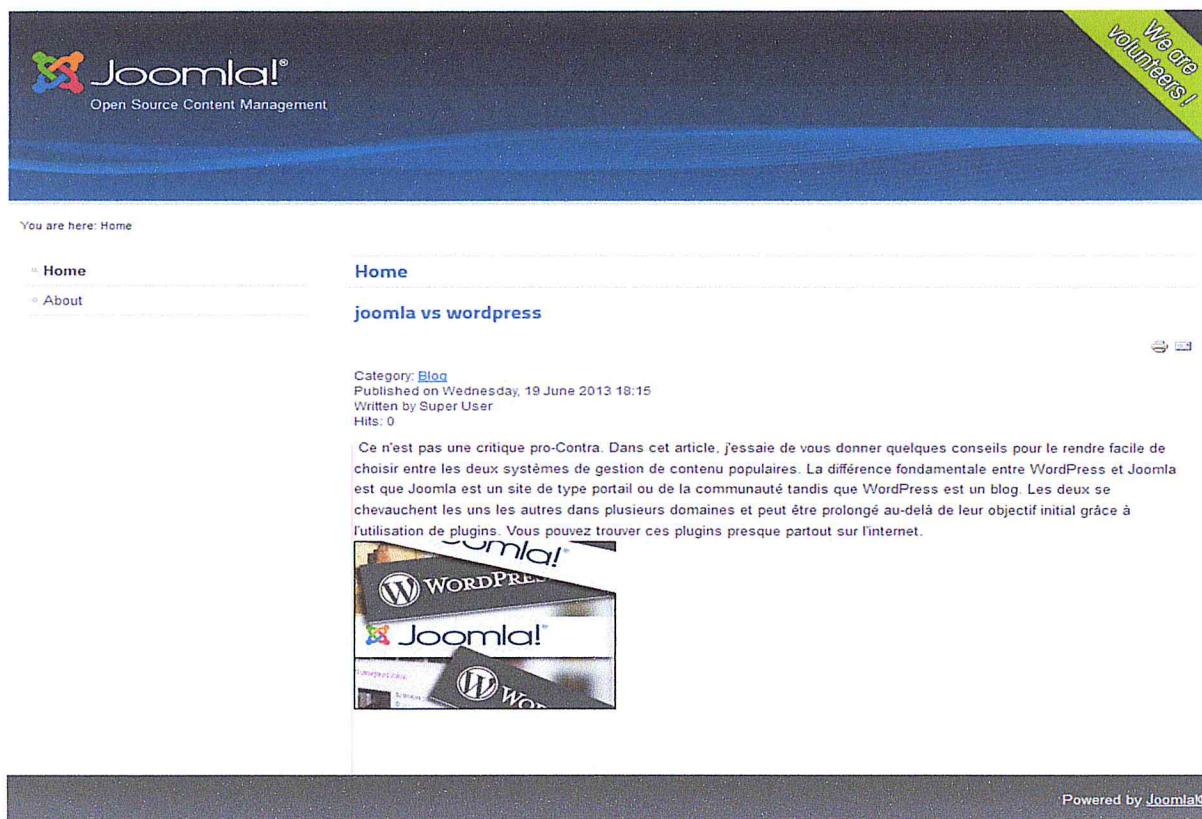


Figure 40 : Article affiché par la plateforme Joomla !

➤ Après l'importation le fichier exporté, on a obtenu le résultat suivant :

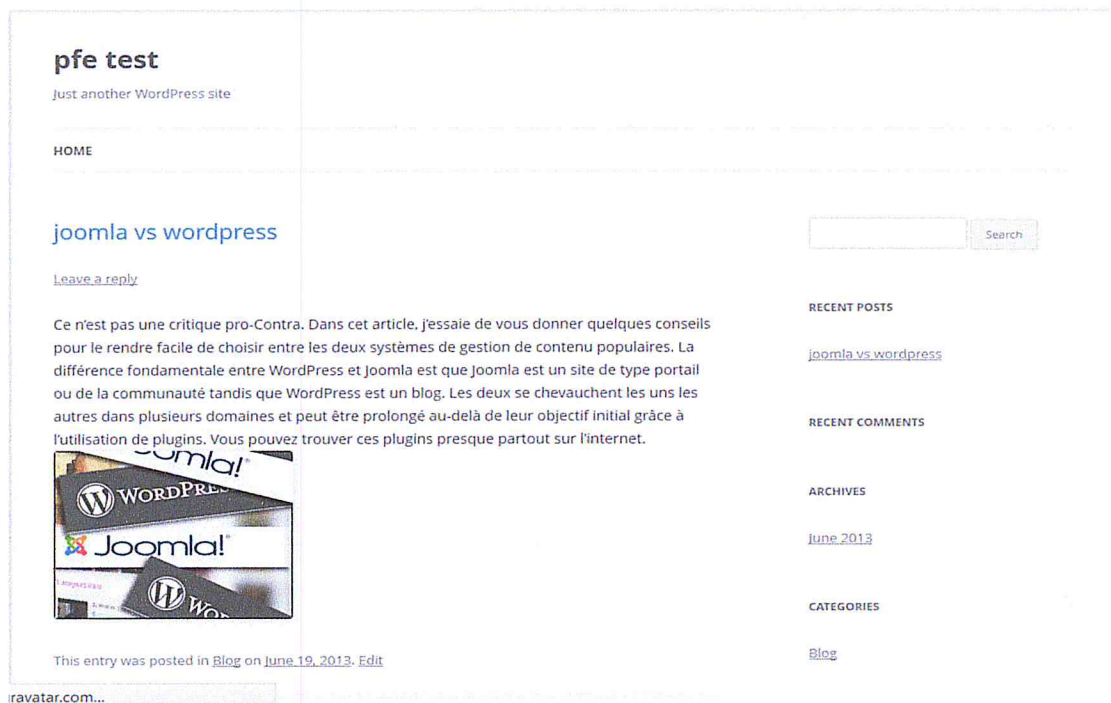


Figure 41 : résultat d'exportation

### **5.3. Discussion :**

Les résultats obtenus montrent l'efficacité des deux approches de migration des données et qu'il n'y a pas de travail manuel requis pour transformer son site du Joomla ! à WordPress.

### **5.4. Conclusion :**

On a présenté dans ce chapitre notre contribution en essayant de répondre aux besoins des utilisateurs liés à la migration des données du Joomla ! vers WordPress, et proposé deux approches de la migration des données : la première est via base de données et la deuxième via un fichier XML, et on a développé les différentes étapes du processus de la migration automatique des données pour l'obtention des résultats fortement satisfaisantes.

# Conclusion

Joomla reste toujours le refuge quand ça revient à la complexité et l'expérience et malgré sa pratique plateforme, elle n'est pas accessible \_de point de vue complexité\_ à tout le monde et exige beaucoup de connaissances. En revanche, WordPress comme on l'a constaté est plus simple et moins lourd ; il est Moins performant pour certaines fonctionnalités et pas forcément adapté à certaines circonstances, mais plus pratique pour un site basic.

C'est exactement cela qui a rendu favorable, l'idée de la migration d'un site Joomla vers un site wordpress, l'idée qui a été éclaircie et mise au point au cours de cette étude.

L'idée elle-même est un grand pas pour les développeurs qui, avant, trouvaient beaucoup de difficultés pour la raison qu'ils devaient tout refaire alors que maintenant ça ne fait pas seulement gagner du temps mais permet de bénéficier du pouvoir \_si on peut dire\_ de wordpress.

# Références

## Bibliographiques

- 1- [Dex\_01] : (Dexter.M, Landry.L, 2012). « Joomla mvc ».
- 2- [Jom\_01] : « MVC » <Joomla.org/mvc>, Mars 2013.
- 3- [Lar\_01] : «Expression régulière » (Largeron.M, 2012).
- 4- [Paro\_01]: (Parolisi.M, 2010) “Joomla! 2.5 overview”.
- 5- [BAD\_01] : Stéphane Badreau. Cristophe Delauve« La gestion de contenu selon ONEXT », 24 Septembre 2004.
- 6- [WEB\_01] : «Système de gestion de contenu »<<http://www.domainepublic.net/-Systemes-de-Gestion-de-Contenu-CMS-.html>>17 mars 2013.
- 7- [WEB\_02] : Dominic Forest, Ph.D. École de bibliothéconomie et des sciences de l'information Université de Montréal INU1050, « Diffusion de l'information numérique Cours 11 »,  
<[http://www.dominicforest.name/documents/INU1050\\_11.pdf](http://www.dominicforest.name/documents/INU1050_11.pdf)> 17 mars 2013.
- 8- [WEB\_03] : « Les systèmes de gestion de contenu »,  
<<http://t3studio.org/index.php?id=196>> 17 mars 2013.
- 9- [WEB\_04]: <<http://livedocs.adobe.com> > 18.
- 10- [WEB\_05]: <<http://livedocs.adobe.com>> 19.
- 11- [WEB\_06] : < <http://blog.canal-web.fr/geoffray/contenu-web/wordpress-drupal-joomla-comparatif-cms-site-contenu/>> 03 mars 2013.

12- [WEB07] : «WordPress, Drupal, Joomla : Quel CMS choisir pour un site de contenu »  
<<http://blog.canal-web.fr/geoffray/contenu-web/wordpress-drupal-joomla-comparatif-cms-site-contenu/>>.

13- [WEB\_08]: Johan Janssens, «Joomla Framework»,  
<<http://www.slideshare.net/JohanJanssens/phpbootcamp-joomla-framework>>,  
Septembre 2006.

14- [wiki\_01]: “Reverse engineering” <[Wikipedia.org/reverse\\_engineering](http://Wikipedia.org/reverse_engineering)>.

15- [wiki\_02]: « MySQL workbench » <[wikipedia.org/mysql\\_workbench](http://wikipedia.org/mysql_workbench)>.

16- [wiki\_03]: “mvc joomla” <[Wikipedia.org/joomla\\_mvc](http://Wikipedia.org/joomla_mvc)>.

