Comparatif de serveurs de contenus

(CMS – Content Management Server)

Jean Xech
Service de la Communication - Université de Perpignan
Jean.Xech@Univ-perp.fr

Albertine Rabat
Service de la Communication - Université de Perpignan
Albertine.Rabat@univ-perp.fr

Aline Savarèse
Service de la Communication - Université de Perpignan
Aline.Savarese@univ-perp.fr

Résumé

Les CMS sont des systèmes de gestion de sites web qui permettent de séparer la gestion des contenus de la présentation à travers des gabarits. Il en existe un très grand nombre de qualités différents avec des technologies diverses. Nous présenterons et détaillerons les critères de choix qualitatifs d'un CMS pour un site d'établissement ou d'équipe que nous appliquerons à ceux choisis pour cet article. En conclusion nous évoquerons le futur possible de ces CMS.

1. Introduction

Les premiers sites web ont été édités par des équipes de passionnés ayant à la fois des connaissances techniques et éditoriales pour créer des pages : connaissances techniques pour les mises à jour et éditoriales en utilisant majoritairement des éditeurs de texte simples ou HTML. Cependant la gestion de pages statiques pose rapidement de nombreux problèmes techniques dès que le site devient important et que le nombre de rédacteurs augmente.

Afin de solutionner ces problèmes, des logiciels spécifiques ont été créés : les serveurs de gestion de contenus (CMS en anglais). Ce sont des logiciels qui vont permettre de répondre aux problèmes qui se posent lors de la création et de la maintenance d'un site web de taille importante comme un site institutionnel ou d'établissement. En particulier :

- définition et gestion des intervenants : les CMS ont permis de bien définir les fonctions des intervenants comme administrateur de site, de sous-site, validateur et contributeur ou rédacteur, ainsi que leurs droits de création (ou de mise à jour, ou de publication) avec leur portée sur le site ou partie de site ;
- gestion du cycle de vie de l'information : suivi des versions, validation, retour en arrière, durée de vie, archivage automatique ;
- mise à jour du site : les interfaces de mise à jour doivent permettre à n'importe quel agent d'une structure de mettre en ligne de l'information sans difficulté technique et selon les droits accordés. On va retrouver pour aider les rédacteurs des assistants, des formulaires et des éditeurs HTML en ligne ;
- l'unicité de mise à jour : le CMS va permettre (par son stockage des informations dans une base de données) de permettre à une information saisie une fois d'être publiée en plusieurs endroits ;
- la conception graphique (design) du site : la présentation est automatisée ce qui uniformise la présentation par l'utilisation de gabarits et des principes de navigation (construction automatique de menus) tout au long du site. Le concepteur de site peut changer très facilement tout élément graphique ou ergonomique grâce au principe des gabarits.

Les CMS ont permis d'industrialiser la mise en œuvre de la gestion de contenu de sites web ayant des contraintes fortes : forte audience, mise à jour fréquente du contenu, sécurité basée sur les droits donnés à des rôles, temps réduit de téléchargement des pages, contenu multimédia, transactions commerciales. Les premiers domaines d'application ont été les sites éditoriaux (informations de natures diverses), les communautés en ligne (internautes partageant des centres d'intérêt d'ordre général ou professionnel), les bases de connaissances (applications intranet ou extranet permettant de capitaliser l'information et le savoir-faire au sein d'une entreprise).

2. Définition du problème

Il existe de nombreuses solutions de gestion de contenus qui possèdent des fonctionnalités diverses et variées

allant de la gestion simple de contenus (ils affichent des pages selon une structure hiérarchique) au système collaboratif (un partage de documents avec une gestion de groupes). Le site de Wikipédia [1] concernant les CMS cite plus de 120 logiciels ; une recherche sur le site « freshmeat.net » donne plus de 190 projets et sur « sourceforge.net » on trouve 1875 projets en mars 2007. Le site cmsmatrix.org [2] propose de comparer (sous forme d'un tableau) jusqu'à 10 CMS parmi 762. C'est justement ce grand nombre de solutions dont on se doute bien qu'elles ne sont pas toutes égales (en termes de qualité et de performance) qui conduit à une difficulté de choix.

Pour la réalisation du site d'établissement ou d'une équipe on va donc retrouver la problématique qui a conduit à la création des CMS : un grand nombre de rubriques, une complexité de navigation, une présentation qui doit assurer une homogénéité et respecter des variantes, le site web pouvant s'adosser au système d'information de l'établissement (c.à.d. aux logiciels métier comme Apogée, l'annuaire LDAP) dans une perspective de communication.

Il faut donc d'abord définir ce qu'est la complexité d'un site pour ce qui concerne ses principales catégories d'organisation (les rubriques et la navigation (le « rubricage »), la présentation, les contributeurs et le back-office, les connexions nécessaires avec le système d'information de l'établissement). Enfin avant de parler de choix il faut bien définir le rôle de chacun des intervenants qui seront : la direction, le service de la communication, le service informatique, les représentants d'entités reconnus au sein de l'établissement.

3. Choisir un CMS, les critères

Il existe plusieurs sites présentant des comparatifs de CMS. Le plus complet (cmsmatrix.org) recense 762 CMS, la comparaison s'effectuant par un tableau comprenant 10 grandes catégories, chacune étant détaillée. Ce comparatif est purement descriptif sans aucune évaluation qualitative. Pour notre part, nous avons regroupé les critères d'évaluation de CMS sous quatre rubriques principales :

- **la description du logiciel** : la technologie, la plateforme système, le serveur Web (IIS ou Apache), la base de données, l'API, l'installation, la maintenance, la documentation, le marketing, les performances, la gestion multi site et multi langues ;
- **le management du CMS et la création de contenu** (le back-office) : la gestion des rédacteurs, des droits, la création des contenus, l'administration technique du CMS. On notera l'aide apportée à la saisie des contenus, en particulier la présence ou non d'un éditeur de texte en ligne intégré ;
- **la présentation** (front-office) : séparation contenu/présentation, création d'interfaces multiples, types de navigation, présentation gérée par XHTML/CSS, intégration du webdesign web 2.0, les bannières multiples ;
- **les modules intégrés et les extensions** : forums, enquêtes et votes, gestion et publication d'événements (par calendrier), bibliothèques de documents, de photos ou de vidéos, gestion automatique de publipostage ou de lettres d'information etc.

Nous retiendrons principalement les critères deux et trois dans notre comparatif. L'étude présentera chacun des CMS selon le plan suivant :

- pré-requis techniques ;
- présentation des concepts du logiciel et sa technologie, sa mise en œuvre et sa maintenance (prêt à l'emploi, prise en main technique et organisationnelle) ;
- administration du CMS et du site, le back-office à destination des webmasters et des rédacteurs ;
- plasticité (souplesse) : à la fois pour l'organisation ou réorganisation du site et de ses rubriques, et la construction de la présentation par gabarit ;
- performances et les modules annexes (gestion de cache, indexation et statistiques d'accès) ;
- une conclusion sur le CMS et son usabilité.

Ce comparatif devra permettre un choix en connaissance de cause sur les avantages et inconvénients de chacun de ces CMS. Il n'y aura donc aucun classement.

4. Les CMS comparés : leur choix

Les CMS retenus seront évalués pour la construction d'un site d'établissement ou d'équipe, et à ce titre ils doivent :

- être téléchargeables et prêts à l'emploi ;
- avoir de la plasticité pour l'organisation/réorganisation du site et de la présentation (possibilité de création de gabarits par un utilisateur averti sans programmation ni compilation ni recours à une société de service …) ;
- leur administration et leur back-office devaient pouvoir être pris en main par un utilisateur averti avec une gestion de cycle de vie des contenus (workflow) ;
- les modules et à minima : présentation de contenus (textes, images et vidéos mixés), bibliothèque de documents, bibliothèque d'images, formulaires (un « mail form »), moteur de recherche (sur texte et méta données), gestion d'événements (calendrier), gestion de « newsletter » ;
- optionnel : forum, blog, intranet, multi sites et multi langues, statistiques, site de commerce.

L'essentiel de notre évaluation sera qualitative. Nous utiliserons à ce propos le concept d'usabilité qui mesure à

quel point un produit, un système, un service d'information est prêt à l'usage. Dans notre cas l'usabilité sera mesurée par le degré de facilité d'apprentissage du logiciel, de la mémorisation des commandes (des interfaces et de leur ergonomie) et de leur utilisation (courante ou épisodique).

Sur le plan technique, nous avons retenu plusieurs technologies en cherchant à couvrir tous les champs possibles. Tous les CMS testés devaient être dynamiques avec des contenus stockés dans une base de données, les pages étant construites à la volée.

Compte tenu du très grand nombre de solutions, notre choix a une part arbitraire. Une première phase de test rapide a permis d'éliminer un grand nombre de CMS, ne conservant que les CMS qui s'installaient sans erreurs avec une présentation par défaut (les promoteurs de PLUME ont procédé de la même manière [3]). Les CMS retenus seront SPIP, Joomla, Drupal et Zope. Ces divers CMS sont construits sur des concepts différents avec des technologies différentes : Python pour Zope, PHP pour Spip, Joomla et Drupal.

Ces CMS sont proposés en chargement, prêts à l'emploi (moyennant l'adaptation nécessaire de la présentation), pour certains très utilisés dans le monde académique et pour d'autres présentant une technologie originale. Le même site exemple sera réalisé avec chacun d'eux.

5. Les tests sur prototype

Il est difficile de concevoir un site prototype qui en tout état de cause ne pourra présenter toutes les caractéristiques de la complexité d'un site réel. Pour les tests et afin malgré tout de tester la programmation de la réalisation des gabarits, nous avons utilisé les pages d'un site de laboratoire et plus précisément celui du CERTAP (Centre d'Etudes et de Recherche juridiques sur les Transformations de l'Action Publique, EA 3682). Ce site est entièrement réalisé en HTML et s'appuie sur une feuille de style.

Le **contenu** : il est constitué pour l'essentiel de pages, certaines contenant des photos. Nous avons rajouté une « visite virtuelle » du laboratoire sous la forme d'une bibliothèque d'images (du site de Perpignan et de Narbonne). Les colloques seront annoncés dans une page d'événements (avec un calendrier quand c'est possible). Enfin nous avons ajouté une page contenant un formulaire de demande d'une copie d'article, la demande étant envoyée par messagerie à l'administrateur du site.

La **conception graphique du site** : toute la conception graphique du site est contenue dans une feuille de style (certap.css) qui sera utilisée avec chacun des CMS comparés. La feuille de style définit l'essentiel de la présentation typographique du site.

La **page d'accueil** du site prototype a un format unique et classique : le logo du laboratoire en haut et à gauche, un menu horizontal présentant les rubriques principales.

Les **pages des rubriques** se présentent sous la forme classique : bandeau en haut, menu à gauche, pied de page et le contenu dans la partie centrale. La construction du menu sur la gauche doit être automatique ; sous le titre de la rubrique on doit retrouver les sous-rubriques ou la liste des articles (rubriques et articles sont des concepts utilisés par Spip). On part du principe selon lequel chaque rubrique a un gabarit qui diffère simplement par la couleur du fond.

6. SPIP

SPIP [4] est un système de publication pour l'Internet qui s'attache particulièrement au fonctionnement collectif, au multilinguisme et à la facilité d'emploi. C'est un logiciel libre, distribué sous la licence GNU/GPL. Le programme est né en 2001 d'une initiative du *minirézo*, un collectif défendant le Web indépendant et la liberté d'expression sur Internet.

Pré-requis techniques

La dernière version de SPIP (à la date de rédaction) est la version 1.9 qui comprend des changements majeurs dans l'organisation technique de SPIP (répertoires et fonctionnement). SPIP fonctionne avec PHP (4 ou 5) et MySQL en environnement Linux/Apache ou Windows/IIS. L'installation et le paramétrage de SPIP se font depuis une interface web (il faut des droits de création/modification pour MySQL).

Il est possible de cloner la version installée ce qui permet d'éviter de dupliquer le code (on peut dire qu'il est multi-site). Par conception SPIP est multilingue et il peut gérer (de nombreuses façons) les sites multilingues : dans la configuration du site, à la section « langues » il faut activer la gestion du multilingue, et choisir les langues qui seront utilisées sur le site.

Concepts

SPIP est un CMS destiné à la publication de revues électroniques. Il propose comme concept de base les notions (communes aux publications papier) de rubrique, sous-rubrique, article, brève et mots clés. Chacun de ces éléments constitutifs du CMS sont fortement structurés et non modifiables :

- la rubrique : titre, descriptif rapide, texte explicatif ;
- le mot clé : Nom (ou titre), le groupe, un descriptif rapide, le texte explicatif, le logo (+ le logo pour le survol) ;
- l'article : le titre, la date de création, les auteurs, le contenu, les documents joints, le logo, le forum associé ;
- l'auteur : signature, courriel, biographie, site web (et URL) ;

– la brève : en complément des articles, SPIP intègre un système de brèves, de courtes notes d'information comme celles utilisées par des revues de presse (ou des revues de Web).

Le cycle de vie d'un article est limité et ne peut être modifié ou étendu : le statut d'un article pourra être *en cours de rédaction – proposé à l'évaluation, publié en ligne, refusé* (et à la *poubelle* c.à.d. irrécupérable).

L'administration du CMS et du site (back office)

L'interface d'administration/création/modification du site se présente sous la forme d'une page web avec un ensemble d'outils (sous forme d'icônes).

L'administration du site, gestion des utilisateurs : les droits se résument à administrateur ou rédacteur. On peut attribuer ces rôles à des auteurs sur la hiérarchie du site (par rubriques).

La création d'une rubrique : les rubriques sont numérotées dans l'ordre de création. La hiérarchie du site se présente sous la forme de rubriques emboîtées, la première (non accessible) étant la racine du site. On peut déplacer les rubriques.

La création d'un article : un éditeur en ligne est disponible avec une barre d'outils spécifique à SPIP (qui utilise des abréviations particulière pour les attributs graphiques du texte). La mise en page du texte est sommaire ; aucun outil pour la création de tableau ou l'insertion d'un lien. L'insertion d'images n'est pas naturelle (pas de glisser/déposer). On peut joindre des documents (images, texte ou autre).

Le design graphique du site

SPIP est fourni avec un ensemble de gabarits qui permettent de démarrer un site par défaut. Les fichiers gabarits sont des fichiers HTML qu'il faut modifier. La présentation d'une page est construite en HTML en s'appuyant sur les feuilles de style fournies dans la distribution SPIP (que l'on peut modifier) et y adjoindre si nécessaire des JavaScripts ; une fois la présentation réalisée, on va y ajouter le code SPIP (sous forme de boucles et de références aux attributs des objets à présenter : titre/auteur/contenu/date) pour créer la partie dynamique de la page.

Exemple de boucle présentant la liste des titres de tous les articles contenus dans la rubrique en cours :

```
<BOUCLE_rubriques(RUBRIQUES){id_rubrique}>
<BOUCLE_articles(ARTICLES){id_rubrique}>
<span class=titre_article>#TITRE</span>
</BOUCLE_articles>
</BOUCLE_rubriques>
```

Il reste à apprendre la syntaxe d'écriture des boucles SPIP et des règles de sélection ou de présentation (conditions et/ou ordre de présentation). On ne peut contrôler l'affichage de listes construites d'articles ou de rubriques que par le nom (en liste alphabétique) ou par le numéro de création, ce qui est très restrictif.

SPIP applique un traitement typographique à tous les textes tirés de la base de données. En particulier, il place des espaces insécables avant certains symboles (point-virgule, point d'interrogation, etc.), et analyse des raccourcis de mise en page. On peut court-circuiter ce traitement pour appliquer ses propres styles.

On peut construire un gabarit spécifique à une rubrique avec héritage (de la présentation) pour les articles et rubriques contenus.

En conclusion pour créer un site personnalisé il suffit de modifier à minima trois fichiers : index.html, rubrique.html et article.html.

La bibliothèque d'images est intégrée.

Réalisation du prototype

Le prototype a pu être réalisé avec :

– le gabarit de page d'accueil ;

– un gabarit par rubrique (seul le fond change en utilisant les CSS) :

– la page de bibliothèque d'images a été réalisée (sans effet de type pellicule photo) en utilisant la méthode préconisée par SPIP ;

– formulaire de recherche (module interne).

Mettre en ligne un agenda demande de la programmation.

Les performances, les modules annexes

On peut vider le cache contenant les pages construites et réinitialiser l'indexation du site (réalisée par le moteur de recherche interne). On trouvera également des statistiques d'accès.

Les commandes de sauvegardes et restaurations de la base de données sont intégrées à l'administration du site. Les sauvegardes sont des exportations du contenu de la base de données seulement sous forme d'un fichier XML qui peut être compressé. Ces procédures nécessitent d'être accompagnées de vraies procédures de sauvegarde.

Une conclusion

SPIP s'installe en moins d'une heure et peut être utilisé avec les paramètres par défaut. L'interface d'administration du site est très simple. La création d'une hiérarchie est facile, intuitive et modifiable.

La création de modèles de pages nécessite une expertise minimale en programmation : connaissance du HTML, des feuilles de style CSS et de la logique d'extraction des données de la base de données SPIP (syntaxe de boucles avec paramètres).

La création d'un texte demande un apprentissage non complexe. Il faut regretter l'utilisation d'abréviations SPIP en lieu et place de la syntaxe HTML et l'absence

d’un vrai éditeur en ligne. L’usage du copier-coller est possible (y compris le HTML pour les experts).

SPIP est conçu principalement pour la publication de textes (d’articles) regroupés par rubriques ou mots clés. On peut facilement l’utiliser pour la création de blogs (des exemples de gabarits spécifiques aux blogs sont proposés sur le site de SPIP). La gestion efficace d’un cache (paramétrable) ne lui interdit pas d’être utilisé pour des sites de grande taille.

7. Joomla

Joomla [5] [6] est la transcription phonétique d'un mot swahili qui signifie « tous ensemble », ou encore « en un tout ».

Il a été créé en 2005 par des développeurs du CMS Mambo. Au départ très semblable à Mambo, Joomla s’en différencie de plus en plus au fur et à mesure des versions.

Pré-requis techniques

La dernière version de Joomla (version 1.5) étant encore en phase de développement, nous avons étudié la dernière version stable (version 1.0.13).

Joomla fonctionne avec PHP et MySQL en environnement Linux/Apache ou Windows/IIS.

L’installation se fait très facilement en quatre étapes depuis une interface web.

Concepts

Joomla dispose de deux sortes de contenus : les contenus statiques et dynamiques.

Les contenus statiques sont surtout utilisés pour des informations ne nécessitant pas d’être modifiées et ne disposent que d’un seul champ de texte. Les contenus dynamiques disposent d’une introduction et d’un texte principal pouvant être caché et accessible via un lien posé sur les mots *Read more*.

De plus, les contenus dynamiques doivent obligatoirement appartenir à une catégorie elle-même incluse dans une section.

On peut ensuite créer des menus permettant d’accéder à une catégorie ou à une section complète.

Les contenus peuvent être non publiés, publiés ou archivés. On a aussi la possibilité de créer une minuterie pour la publication d’un contenu.

Les éléments supprimés dans Joomla sont redirigés vers la poubelle où ils sont stockés jusqu’à une suppression définitive.

L’administration du CMS et du site (back office)

L’administration du site est accessible à partir de n’importe quel navigateur et se présente sur la forme d’un menu et d’icônes d’accès rapide à des fonctionnalités spécifiques du CMS.

Il est possible de changer le template du back office.

Le tableau des rôles et des droits attribués en standard (groupe *Frontend* : Section Site de Joomla ; groupe *Backend* : Section Administration de Joomla) :

Joomla!	Frontend				Backend		
	Proposer un article	Editer SON article	Editer TOUS les articles	Publier un article	Editer les menus et les articles	Administrer - Composants - Modules - Mambots	Administrer : - Configuration - Langues - Templates - Corbeille - Utilisateurs
Registred	✓	✗	✗	✗	✗	✗	✗
Author	✓	✓	✗	✗	✗	✗	✗
Editor	✓	✓	✓	✗	✗	✗	✗
Publisher	✓	✓	✓	✓	✗	✗	✗
Manager	✓	✓	✓	✓	✓	✗	✗
Admin	✓	✓	✓	✓	✓	✓	✗
SuperAdmin	✓	✓	✓	✓	✓	✓	✓

Référence du tableau : http://www.joomlafacile.com/content/view/242/61/

La gestion des droits n’a pas une granulométrie très fine et il est conseillé d’utiliser des extensions comme GMAccess qui rend possible la création de groupes d'utilisateurs ainsi qu’une gestion centralisée des accès aux différents composants d’un site Joomla (contenus, menus, modules etc.).

Pour la création d’article on peut coder directement en HTML ou utiliser un éditeur wysiwyg comme JCE ou TinyMCE. Le copier/coller nécessite un éditeur wysiwyg.

L’insertion d’image n’est pas intuitive et nécessite l’utilisation d’une application spécifique à Joomla comme JW Media Manager XTD qui permet de naviguer dans tout le site, de manipuler tous les documents stockés et de modifier les images (taille en particulier).

Le design graphique du site

Joomla dispose d’un nombre important de gabarits gratuits disponibles sur Internet que l’on peut télécharger [7] et installer. Il est possible de créer soi-même son modèle personnel de gabarit en créant une archive comportant tous les fichiers nécessaires. A cette fin on peut utiliser le logiciel Dreamweaver qui possède une extension de création de template Joomla ce qui permet d’intégrer facilement au modèle les balises PHP spécifiques aux templates Joomla.

Dans un gabarit, on peut charger des modules constitués de menus à l’aide de balises PHP de positionnement. Par contre la création d’un menu se fait manuellement à travers une interface web en ajoutant des items. Ces items peuvent être des liens vers des articles mais aussi des composants particuliers ou la liste d’une section. Lors de la création d’un menu, un module est automatiquement créé et ainsi on peut l’afficher dans une page à l’aide de sa position dans le *template*.

Les modules peuvent être affichés pour toutes les pages ou être attachés à certaines pages seulement.

Il est possible d’assigner un gabarit (*template*) spécifique à certaines pages du site.

Réalisation du prototype

Pour réaliser le prototype, nous avons créé le gabarit spécifique à la page d'accueil et les gabarits pour les pages intérieures en installant à chaque fois un nouveau dossier zip et en assignant le gabarit aux pages adéquates.

Pour la galerie d'images, le formulaire de recherche ou l'agenda, ce sont des modules intégrés et à installer qui ont été utilisés.

Les performances, les modules annexes

Joomla est un CMS open source qui dispose de nombreux modules supplémentaires : annuaires, agenda, gestion de news, blogs, chat, messagerie, gestionnaire de formulaires, forums, media et galeries photos, publicité et annonces, outils d'administration (plan de site, statistiques).

Conclusion

Joomla est un CMS facile à installer qui ne nécessite pas de connaissances approfondies en programmation PHP pour la création de gabarit.

Cependant, l'administration du site (le back office) n'est pas très intuitive. Il faut un approfondissement pour se familiariser avec les notions de sections, de catégories, de menus et de modules propres à ce CMS.

On ne construit pas le site de façon hiérarchique mais plutôt de façon thématique. Ainsi il n'y a que deux niveaux hiérarchiques section puis catégorie. Cela nécessite donc un travail préalable d'organisation des articles.

Grâce à un important panel de modules et de *templates* présents sur le web, on peut utiliser ce CMS pour de nombreux types de sites.

8. Drupal

Drupal [7] (prononciation à l'anglaise du mot hollandais "druppel" qui veut dire goutte) n'a pas de module d'installation automatique. Elle doit se faire "à la main" : il faut créer la base de données puis renseigner l'interface d'installation.

Drupal a comme fonctions caractéristiques un espace utilisateur (biographie, contact, liste des publications), un espace de création de contenu (workflow, suivi de modification et éditeur WYSIWIG possible), un forum et une galerie d'images intégrés, un système de template en PHP très flexible, la possibilité de recevoir le contenu par email, une administration extrêmement complète (automatisation, cache, url rewrite, etc.) qui peut demander l'ajout de modules complémentaires.

Si l'on veut présenter l'interface d'administration en français (pour les administrateurs et auteurs) il faut installer la version française, activer le module *locale* puis installer la traduction (le téléchargement du fichier fr.po peut poser des problèmes, les traductions ne sont pas complètes pour le noyau ni pour certains modules).

Drupal peut gérer des multi-sites et des sites multilingues.

Concepts

Les contenus stockés dans une base de données sont organisés sous forme de *nodes*, de *users* et de *comments*.

Nodes : De nombreux types de *nodes* sont prédéfinis : il s'agit des *pages* (pages statiques HTML), des *stories* (page web structurée), des *blogs*, des *forums*, des *polls (*sondages*)*. Un *node* standard comporte un titre, un résumé et un corps et il est enregistré dans la base avec des informations relatives à l'auteur, la date de création et son statut. Des procédures adaptées permettent de le manipuler (création, modification, visualisation).

Drupal propose un système de taxonomie qui permet d'associer plusieurs mots-clés aux contenus et de créer une gestion dynamique des catégories avec autant de niveaux que nécessaire. Très puissant, ce système peut permettre de modifier la navigation dans un site.

Un *node* peut être configuré pour accepter en attachement des commentaires sous la forme classique de fils de discussion avec une gestion d'autorisations par groupe. Ces commentaires sont stockés à part du node dans la base de données.

Users : les utilisateurs créés se voient attribuer des rôles qui comportent des droits d'accès.

Les modules **CCK** (Content Construction Kit) et **Views** : CCK permet de créer de nouveaux types de *nodes* dans Drupal. Lors de la création d'un nouveau type de *node* on indique quels sont les champs le constituant et le type de ces champs (texte, date, nombre, email, lien, liste de sélection, référence à un autre *node*...). Ces nouveaux types de *node* peuvent ensuite être configurés comme tout autre type de *node* (*comments*, d'attachement de fichiers, de catégories...). Le module **Views** (compagnon de CCK) permet de générer des blocs ou des pages de listes (résumé ou contenu complet) à partir de tout type de *node* dans Drupal.

Comments : un node peut être configuré pour accepter en attachement des fils de commentaires (comme dans un forum) écrits par des utilisateurs autorisés (par groupes). Les commentaires apparaîtront sous la forme typique d'un topic de forum ou d'une entrée de blog.

A ces concepts de base s'ajoutent le *block* et le *module* :

Blocks : le block est une unité de données construite pour être incluse dans une page web dans n'importe quelle zone (une interface web permet de construire le block et de le placer).

Modules et Hooks : le module est l'unité d'extension de Drupal qui est constitué d'un noyau non modifiable. Chaque module peut définir ses propres *nodes* et c'est un mécanisme de fonctions prédéfinies, les *hooks* (fonctions à programmer), qui permet au noyau de Drupal d'exécuter le module.

L'administration du CMS et du site (le back-office)

L'interface d'administration comprend quatre grandes catégories et un service de consultation de logs : gestion du contenu - gestion des utilisateurs - construction du site – configuration du site – les logs.

Gestion des utilisateurs : la notion de rôle permet de centraliser la gestion des droits. Deux rôles sont déclarés par défaut : anonyme et authentifié. Comme la liste des permissions est très détaillée on peut hiérarchiser les droits donnés. Il faut commencer par définir les rôles puis déclarer les utilisateurs.

Par défaut Drupal accepte les demandes de création de compte (le mot de passe initial est envoyé au courriel donné) ce que l'on doit modifier dans les *users settings* du panneau d'administration.

Configuration du site : on trouvera de manière classique un certain nombre de paramètres à définir comme l'emplacement des fichiers téléchargés.

Construction du site : on trouvera les interfaces d'activation des modules installés, de construction de blocks et des menus, de la gestion des thèmes.

Gestion du contenu : elle comprend les catégories, commentaires, contenus, types de contenus, forums. Les catégories peuvent être hiérarchiques multiples ou simples ce qui ouvre la taxinomie à de la complexité.

La création d'une page dynamique (*story* : elle sera attachée à une hiérarchie du site (*Parent item*)) se fait à travers un formulaire sans fioritures ni éditeur permettant d'enrichir le texte (de gras, italique, lien ou tout autre élément classique HTML). On a la possibilité de visualiser le texte (*preview*) avant sa publication. Aucune indication n'est donnée pour l'insertion d'une image (il faut attacher le fichier puis inclure le lien sous sa forme HTML …).

Types de contenus : quelques paramètres sont à voir, en particulier l'autorisation de téléchargement de documents (dont la taille maximale peut être fixée).

La gestion du cycle de vie du document : le workflow est paramétrable en utilisant les modules *workflow* et *actions*. Le module *workflow_access* complète le processus en permettant de gérer de façon très fine les droits de chaque rôle en fonction de l'état dans lequel se trouve l'article.

Le design graphique du site

Drupal sépare le contenu de la présentation en utilisant un système de thèmes avec trois moteurs de rendu : PHPTemplate, XTemplate, Smarty (une autre version de PHPTemplate).

PHPTemplate utilise des gabarits qui sont autant de bouts de code PHP, du code xHTML (s'appuyant sur des feuilles de style) et incluant les variables représentant les contenus à présenter. On retrouve un procédé ressemblant à celui de SPIP. Les thèmes peuvent être attachés aux modules. Pour créer son thème il suffit d'en dupliquer un puis de modifier les gabarits de base *block.tpl.php*, *page.tpl.php*, *box.tpl.php*, *node.tpl.php* et *comment.tpl.php*, de remplacer le logo et le fichier *style.css*.

La base de la présentation est le *block* que l'on peut positionner dans la page. C'est à partir de l'agrégation de *blocks* de base fournis par chaque module que l'on construire les *blocks* constituant la page.

Par défaut, Drupal construit les pages suivant une structure constituée d'une colonne centrale (le contenu), deux panneaux (*sidebars*) à droite et à gauche et deux zones « haut de page » et « bas de page ».

En bref la personnalisation est une construction méthodique dont la logique n'est pas évidente.

Réalisation du prototype

Le prototype a pu être réalisé en utilisant le moteur de rendu PHPTemplate :

- le gabarit de page d'accueil ;
- un seul gabarit de rubrique ;
- la page de bibliothèque d'images a été réalisée en utilisant le module *AcidFree Album* (installation du module et configuration) ;
- formulaire de recherche : il faut utiliser le module Forms et son API pour construire le formulaire (non réalisé pour l'article) ;
- agenda : on a utilisé le module *event* (dont les champs peuvent être personnalisés avec le moteur CCK (le Content Construction Kit permet de rajouter des champs à tout *node* défini et de les utiliser dans le rendu).

L'ensemble de la réalisation a demandé plusieurs jours de travail (juste pour une évaluation).

Les performances et les modules annexes (gestion de cache, indexation et statistiques d'accès)

On trouvera sur le site de Drupal un très grand nombre de modules complémentaires et complets : formulaire, newsletter, messagerie, chat, enquête ...

Pour chacune des catégories de module on trouve un nombre élevé de solutions : administration (133), CCK (83), community (101), content (226), content display (252), developer (77), e-Commerce module (14), evaluation/rating (40), event (24), file management (30), mail (56) , media (85), multilingual (12), taxonomy (81), theme related (63), user access/authentication (90), user management (60) …

Un système de cache permet une gestion efficace des pages destinées à des utilisateurs anonymes. Un chronographe (*Cron*) permet de programmer des opérations, comme la réalisation de statistiques.

Les statistiques accessibles depuis l'interface d'administration concernent quelques « logs » : les dernières connexions, les accès récents refusés, les dernières pages non trouvées.

Les sauvegardes/restaurations du site doivent être réalisées comme des opérations système (en utilisant des procédures pour la base de données et les répertoires système de Drupal).

Conclusion

Drupal possède bien des qualités : une structure de données évolutive et adaptable, un système de rendu programmable, une interface de gestion complète et très élaborée.

En contrepartie de ces qualités, on trouvera de la complexité pour son utilisation et sa personnalisation. La création de gabarits avec PHPTemplate demande de connaître la programmation en PHP (on ne peut pas utiliser d'outil de déverminage qui permettrait de retrouver par exemple des chaînes non fermées …).

Il constitue le bon compromis entre un CMS prêt à l'emploi avec des structures de données à publier figées et le CMS en Kit (*framework*). Il ne peut être installé, personnalisé et utilisé sans lire la documentation en ligne ce qui conduit à une prise en main assez longue par les administrateurs du site. Par contre l'interface de mise à jour des contenus est facile à utiliser.

Globalement Drupal est un CMS demandant des connaissances système et de programmation PHP.

9. Zope

Zope [9] est un serveur d'application web orienté objet libre écrit dans le langage de programmation *Python*. Il peut être entièrement géré à partir d'une interface Web. Zope publie sur le réseau des objets *Python* qui sont enregistrés dans une base de données objet, ZODB [10]. Des types d'objets de base, tels que les documents, les images, les gabarits (*templates*) de page, sont à la disposition des utilisateurs pour être créés et gérés via l'internet. Des types d'objets spécialisés, tels que les wikis, les blogs, les galeries de photos, sont disponibles en tant que greffons tiers (appelés dans la terminologie Zope *produits*).

Pré requis techniques

Zope ne nécessite aucun programme ou Framework pour son fonctionnement. Sous Windows, il suffit de télécharger le binaire de la version stable, d'exécuter le setup.exe et de se laisser guider.

Sous Unix, on peut utiliser le format rpm ou tar.gz pour installer Zope. Le compilateur Python est intégré.

Zope peut ensuite être démarré en tant que service. Il est par défaut attaché au port 8080.

Concepts

Zope ne contient pas des pages comme c'est le cas habituellement avec d'autres systèmes de serveurs web mais des objets dans une base de données objet spécifique appelée Zope Object Database (ZODB). Ce sont ces objets qui sont publiés sur le web.

Cette approche permet d'exploiter les avantages des technologies objet, tels que l'encapsulation et l'héritage.

Ces objets possèdent également des méthodes qui permettent leur représentation en HTML. Un objet de classe *folder* est un conteneur d'objets comme un répertoire. Zope relie les URL aux objets en utilisant la hiérarchie de contenu des objets.

Une caractéristique particulièrement innovante de Zope est son emploi récursif de l'acquisition. Ainsi lorsque le CMS ne trouve pas l'objet dans un répertoire, la procédure d'acquisition permet de remonter la hiérarchie jusqu'à trouver l'objet demandé. Il est donc facile de réaliser un gabarit pour une partie du site. L'acquisition « remonte » les conteneurs des objets mais ne les redescend jamais.

La finalité de Zope est la création de sites web dynamiques mais sa philosophie orientée objet (plutôt que pages web) demande une certaine phase d'apprentissage qui à terme permet de construire des sites complexes avec beaucoup d'économie (on dira alors que l'on est « Zope Zen » pour avoir parfaitement compris le mode de fonctionnement de Zope). Cependant et au-delà de ses concepts objets, Zope permet de construire des sites hiérarchiques classiques.

Par ailleurs Zope peut être utilisé comme serveur ftp, xml-rpc, webdav et peut facilement être utilisé avec un serveur web frontal, comme Apache.

L'administration du CMS et du site (le back office)

L'interface graphique de Zope est très intuitive puisqu'elle se présente comme l'explorateur Windows. Les dossiers sont affichés à gauche et leur contenu à droite. De plus, dans la partie de droite on trouve aussi un système d'onglets permettant d'exécuter des actions sur les objets listés.

Les droits des utilisateurs sont administrés selon des rôles. Il existe trois rôles particuliers, *Manager*, *Anonymous* et *Owner* mais il est possible également de créer des rôles supplémentaires. Il est possible d'assigner des droits particuliers à seulement certaines parties du site.

On peut créer une arborescence hiérarchique classique de site web en ajoutant un objet *folder* à l'emplacement souhaité. Dans chaque dossier, on peut ensuite ajouter des objets de plusieurs types, dont des images et surtout des DTML document qui permettent de créer des gabarits de pages dans chaque partie du site.

Comme il n'existe pas dans Zope d'éditeurs wysiwyg, il est donc nécessaire de connaître le langage HTML et les

balises DTML (Dynamic Template Markup Language) qui facilitent l'accès aux objets des dossiers pour créer des pages complètes.

Le design graphique du site

Zope fournit plusieurs mécanismes pour appliquer des patrons à du HTML : le DTML (Dynamic Template Markup Language, Langage de Balisage de Patron Dynamique) et ZPT (Zope Page Templates, Patrons de Pages de Zope). Le DTML est un langage de balisage permettant d'implanter des scripts simples dans les patrons. Le DTML permet l'inclusion de variables, de conditions et de boucles. Comme pour SPIP, le DTML a des inconvénients majeurs : les marqueurs DTML mélangés au HTML forment des documents HTML non valides, et l'inclusion inattentive de logique dans les patrons produit du code illisible. ZPT est une technologie qui résout ces problèmes. Les patrons ZPT peuvent être soit des documents XML ou HTML bien formés, dans lesquels tout le marquage spécial se présente sous forme d'attributs dans le domaine de nom TAL (Template Attribute Language, Langage d'Attributs de Patrons). ZPT offre un ensemble limité d'outils pour l'inclusion conditionnelle ou répétitive d'éléments XML, ainsi les patrons sont habituellement assez simples, avec la majorité de la logique implantée en Python. Un avantage non négligeable des patrons ZPT est qu'ils peuvent être édités dans la plupart des éditeurs HTML graphiques. ZPT offre également le support direct de l'internationalisation.

Réalisation du prototype

Pour la réalisation des gabarits, Nous avons utilisé les balises DTML plutôt que les patrons ZPT qui nécessitent une phase d'apprentissage plus importante.

Un gabarit pour la page d'accueil a été créé à la racine du site puis les autres gabarits dans chaque dossier.

Pour la page de bibliothèque d'images, le formulaire de recherche ou l'agenda, on peut utiliser des produits supplémentaires à intégrer à Zope.

Les performances, les modules annexes

Zope est intéressant par le fait qu'il ne nécessite aucun autre programme pour fonctionner et qu'il a amené une idée novatrice dans la conception web avec la notion d'objet.

Conclusion

Zope est un CMS facile à installer.

Le back office est très intuitif et donc facile d'utilisation. Il faut toutefois un laps de temps avant de se familiariser avec les notions d'objet, d'acquisition, le langage de balisage DTML ou le ZPT. Mais une fois ces notions assimilées, la réalisation des gabarits et des pages est très rapide et efficace. Pour rajouter des fonctionnalités supplémentaires au site web ou créer de nouveaux produits, il faut tout de même apprendre le langage Python.

10. Les autres CMS

Un certain nombre de CMS ont été testés et à des titres divers exclus de cette présentation.

EZ_Publish : très simple à installer dans sa version gratuite ; sous Windows il installe PHP et Apache de manière opérationnelle ; cependant si son interface d'administration est simple d'usage, la création de gabarits est complexe à réaliser. Il exige d'utiliser PHP4 et aucune version n'est prévue pour utiliser du PHP5.

DotNetNuke [11] et Rainbow [12] : il était très tentant de tester ces CMS OpenSource sur un serveur Windows. Début 2002, Microsoft fit réaliser par la société Vertigo Sotfware deux applications complètes avec la technologie DotNet : IBuySpy Portal (portail intranet et internet) et Ibuyspy store (gestion d'une e-boutique) en deux versions (VBNet et C#). Ces deux versions ont donné des CMS avec un développement communautaire. La version C# fut reprise par un italien Emmanuele De Andreis, pour donner le CMS Rainbow. La seconde version fut reprise par un Canadien Shaun Walker pour donner un premier CMS appelé IBuySpy Workshop puis devenu DotNetNuk. Ces deux CMS sont livrés sous forme d'un fichier *zip* sans installateur (la compilation est automatique à la première exécution). Leur installation a échoué et nous en sommes restés la.

Chaque fois que l'installation d'un CMS s'est mal réalisée (avec l'installateur fourni ou en suivant les procédures indiquées) ou qu'à l'usage, des bugs bloquants sont apparus nous avons abandonné l'étude. Ce fut le cas pour ceux en technologie Java, en particulier pour Cofax [13] et OpenCMS [14].

11. Conclusion

Pour notre comparatif de CMS nous avons retenus principalement deux critères : l'usabilité de l'interface d'administration du site (le backoffice) et la réalisation de gabarits pour le design graphique d'un site standard.

Dans le tableau comparatif nous avons ajouté deux autres critères : l'installation du logiciel et les modules annexes.

Nous avons négligé la sécurité qui est un sujet à lui seul (il faut nécessairement se rapprocher des RSSI) de même que l'authentification des utilisateurs dont nous savons qu'il est un point essentiel du système d'information universitaire. L'authentification est une part essentielle du code du noyau des CMS ce qui rend difficile toute adaptation.

Pour nos tests (forcément lacunaires) nous avons choisi un site de laboratoire dont la structure est hiérarchique. Cependant d'autres types de sites auraient pu servir de modèle pour le prototype comme par exemple un site de contenus extraits d'une base de données métier dont la caractéristique est d'être « sémantique » et pas hiérarchique. Le choix du CMS s'en trouverait modifié.

Il en est de même pour la construction des gabarits (le design du site) dont le modèle actuel est une page contenant un tableau à trois colonnes avec un bandeau

(en haut) et un pied de page (parler d'un tableau ne préjuge en rien du code HTML utilisé : tableau ou boîtes DIV) . La construction d'un site sur une ontologie (ou une taxonomie complexe) conduira à une interface bien différente. On pourrait prendre l'exemple de la présentation de l'offre de formation d'un établissement : le design du site sera différent si on passe d'une simple page de recherche par formulaire à une navigation (par grades, disciplines, UFR ou instituts).

SPIP sera parfait pour un site dont la structure est hiérarchique, ne sera pas trop évolutive, en utilisant les modules annexes standards et sans avoir à développer un module spécifique.

Drupal et Joomla sont à la frontière du CMS et du framework. Leur prise en main et leur personnalisation demande du temps et un apprentissage assez long mais leur souplesse et la richesse en modules annexe assure une réussite certaine. La possibilité de construire des sites sur des taxonomies permet de d'organiser des parcours complexes et non plus simplement hiérarchiques. Le développement de modules spécifiques est possible et demande une simple connaissance du PHP.

Zope et son extension Plone [15] (non présenté ici) sont séduisants à plus d'un titre. On peut personnaliser le backoffice, l'utilisation du concept objet permet d'étendre les fonctionnalités de Zope et de construire des sites web sur mesure.

Nous retiendrons de cette étude l'apparition, certes modeste encore, des CMS permettant de construire un site web suivant une taxonomie (début du long chemin menant à l'utilisation d'ontologies ?). Dans ce cas on reste dans un domaine hiérarchique mais plus riche. Sûrement une solution d'avenir …

CMS	Installation	Administration du site	Design graphique	Modules annexes
SPIP	*	*	**	****
Joomla	*	**	***	***
Drupal	*	**	****	**
Zope	*	**	***	****

Tableau comparatif des CMS (l'échelle de valeur a de l' facile à **** complexe ou difficile)*

Le site opensourcecms.com [16] permet de tester en ligne (sans installation) de nombreux CMS dont ceux que nous avons présentés.

Références bibliographiques

[1] Wikipédia *Liste de systèmes de gestion de contenu* [en ligne] http://fr.wikipedia.org/wiki/Liste_de_syst%C3%A8mes_de_gestion_de_contenu

[2] *Comparatif de CMS* [en ligne] http://www.cmsmatrix.org

[3] *Description du processus de choix d'un CMS pour le projet PLUME* Pierre-Yves GOSSET (UREC) [en ligne] http://www.urec.cnrs.fr/IMG/pdf/LL.PLUME_Choix_Drupal.pdf

[4] *Spip* (site web) [en ligne] http://www.spip.net

[5] *Joomla* (site web officiel) [en ligne] http://www.joomla.org/

[6] *Joomla* (site web français) [en ligne] http://www.joomlafrance.org

[7] *Site de gabarits Joomla* (site web) [en ligne] http://www.rockettheme.com/

[8] *Drupal* (site web) [en ligne] http://drupal.org/

[9] *Zope* (site web) [en ligne] http://www.zope.org/

[10] *ZODB* [en ligne] http://www.journaldunet.com/developpeur/tutoriel/pyt/040405-zope-zodb.shtml

[11] *DotNetNuke* (site web) [en ligne] http://www.dotnetnuke.com/

[12] *Rainbow (site web)* [en ligne] http://www.rainbowportal.net/

[13] *Cofax* (site web) [en ligne] http://www.cofax.org/content/cofax/home/

[14] *OpenCMS* (site web) [en ligne] http://www.opencms.org

[15] *Plone, un outil de gestion de contenu* - Frédéric Saint-Marcel (INRIA Rhone-Alpes), Philippe Lecler (IRISA/INRIA) JRES 2005 [en ligne] http://2005.jres.org/paper/46.pdf

[16] *Test en ligne de CMS* [en ligne] http://opensourcecms.com