

Livre blanc

Gestion de contenus les solutions open source

Edition 2010

Grégory BECUE

Version 4

Pour plus d'information : www.smile.fr

Tél : 01 41 40 11 00

Mailto : sdcsmile.fr

PRÉAMBULE

Smile

Smile est une société d'ingénieurs experts dans la mise en œuvre de solutions open source et l'intégration de systèmes appuyés sur l'open source. Smile est membre de l'APRIL - l'association pour la promotion et la défense du logiciel libre, du PLOSS – le réseau des entreprises du Logiciel Libre en Ile-de-France et du CNLL – le comité national du logiciel libre.

Smile compte 280 collaborateurs en France, 330 dans le monde (avril 2010), ce qui en fait *la première société en France spécialisée dans l'open source*.

Depuis 2000, environ, Smile mène une action active de veille technologique qui lui permet de découvrir les produits les plus prometteurs de l'open source, de les qualifier et de les évaluer, de manière à proposer à ses clients les produits les plus aboutis, les plus robustes et les plus pérennes.

Cette démarche a donné lieu à toute une gamme de *livres blancs* couvrant différents domaines d'application. La gestion de contenus (2004), les portails (2005), la business intelligence (2006), les frameworks PHP (2007), la virtualisation (2007), et la gestion électronique de documents (2008), ainsi que les PGIs/ERPs (2008). Parmi les ouvrages publiés en 2009, citons également « Les VPN open source », « Firewall et Contrôle de flux open source » et « Middleware orientés messages », dans le cadre de la collection « Système et Infrastructure ».

Chacun de ces ouvrages présente une sélection des meilleures solutions open source dans le domaine considéré, leurs qualités respectives, ainsi que des retours d'expérience opérationnels.

Au fur et à mesure que des solutions open source solides gagnent de nouveaux domaines, Smile sera présent pour proposer à ses clients d'en bénéficier sans risque. Smile apparaît dans le paysage informatique français comme le prestataire intégrateur de choix pour accompagner les plus grandes entreprises et organisations dans l'adoption des meilleures solutions open source.

Ces dernières années, Smile a également étendu la gamme des services proposés. Depuis 2005, un département consulting accompagne nos clients, tant dans les phases d'avant-projet, en recherche de solutions, qu'en accompagnement de projet. Depuis 2000, Smile dispose d'un studio graphique, devenu en 2007 Smile Digital – agence interactive, proposant outre la création graphique, une expertise e-marketing, éditoriale, et interfaces riches. Smile dispose aussi d'une agence spécialisée dans la Tierce Maintenance Applicative, le support et l'exploitation des applications. Enfin, Smile est implanté à Paris, Lyon, Nantes, Bordeaux et Montpellier. Et présent également en Espagne, en Suisse, en Ukraine et au Maroc.

Ce livre blanc

Ce document n'est pas un simple comparatif entre les CMS open source que nous avons sélectionnés. Nous nous efforçons d'en faire aussi une introduction aux concepts fondamentaux des CMS et de leur mise en œuvre. La gestion de contenus n'est pas qu'une affaire d'outils, et avant de choisir un outil, il convient d'avoir bien analysé les finalités de son projet.

C'est pourquoi nous proposons, pour chacun des concepts les plus importants, notre analyse des enjeux et des possibilités qui différencient les outils du marché.

Après cette analyse, nous essayons de positionner chacun des CMS étudiés par rapport aux besoins que nous pouvons avoir.

Quelques références de Smile

Sites Internet

EMI Music, Salon de l'Agriculture, Mazars, Areva, Société Générale, Gîtes de France, Patrice Pichet, Groupama, Eco-Emballage, CFnews, CEA, Prisma Pub, Véolia, NRJ, JCDecaux, 01-Informatique, Spie, PSA, Boiron, Larousse, Dassault-Systèmes, Action Contre la Faim, BNP Paribas, Air Pays de Loire, Forum des Images, IFP, BHV, ZeMedical, Gallimard, Cheval Mag, Afssaps, CNIL...

Portails, Intranets et Systèmes d'Information

HEC, Bouygues Telecom, Prisma, Veolia, Arjowiggins, INA, Primagaz, Croix Rouge, Eurosport, Invivo, Faceo, Château de Versailles, Eurosport, Ipsos, VSC Technologies, Sanef, Explorimmo, Bureau Veritas, Région Centre, Dassault Systèmes, Fondation d'Auteuil, PagesJaunes Annonces...

ECM & gestion documentaire

Primagaz, UCFF, Apave, Géoservices, Renault F1 Team, INRIA, CIDJ, SNCD, Ecureuil Gestion, CS informatique, Serimax, Véolia Propreté, NetasQ, Corep, Packetis, Alstom Power Services, Mazars...

E-Commerce

De Dietrich, Adenclassifieds, Macif, Furet du Nord, Gîtes de France, Camif Collectivité, GPdis, Longchamp, Projectif, ETS, Bain & Spa, Yves Rocher, Bouygues Immobilier, Nestlé, Stanhome, AVF Périmédical, CCI, Pompiers de France, Commissariat à l'Energie Atomique...

ERP & Décisionnel

Groupe Accueil, Anevia, Projectif, Xinek, Companeo, Advans, Point P, Le Figaro, Bureau Veritas, Mindscape, Loyalty Experts, Cecim, Espace Loggia, Nouvelles Frontières, La Poste, HomeCineSolutions, Vocatis, Skyrock, Lafarge, France Domicile, Polyexpert, Cadremploi, Cmonjob, Meilleurmobilier.com...

Infrastructure et Hébergement

Agence Nationale pour les Chèques Vacances, Pierre Audoin Consultants, Rexel, Motor Presse, OSEO, Sport24, Eco-Emballage, Institut Mutualiste Montsouris, ETS, Ionis, Osmoz, SIDEL, Atel Hotels, Cadremploi, SETRAG, Institut Français du Pétrole, Mutualité Française...

Versions

Version 2.1

Téléchargée à plus de 10 000 exemplaires, la version 1 de ce livre blanc remontait à 2004, et avait besoin d'une mise à jour. Cette version 2.1 est une refonte d'envergure.

Tout d'abord, **nous avons éliminé de notre panel la solution Redhat CMS**, et ceci pour plusieurs raisons : Redhat est aujourd'hui devenu Byline, sous la responsabilité du consortium ObjectWeb, mais le transfert est loin d'être finalisé et le développement semble arrêté. De plus, ObjectWeb a déjà une solution de portail et de gestion de contenus, avec l'intégration récente de eXoplatform, projet plus dynamique. Il nous paraît difficile dans ce contexte de croire à la pérennité de cette solution, par ailleurs peu répandue.

En revanche, **nous avons intégré Apache Lenya et InfoGlue**, deux produits en environnement Java, répondant à des besoins bien distincts. Ces deux solutions sont de qualité et ont déjà plusieurs références à leur actif, ce qui permet de compléter l'offre open source J2EE, jusqu'ici relativement réduite.

Enfin, nous avons mis à jour nos commentaires avec les évolutions importantes des différentes solutions. **Typo3 a notamment beaucoup évolué**, avec les extensions TemplaVoila pour les structures de contenus, et les extensions de gestion de versions et de workflow qui lui manquaient tant. **OpenCMS a lui aussi profité** de sa dernière version pour se doter de structuration de contenus, tandis que **eZ publish** apporte un lot impressionnant d'amélioration dans ses versions **3.6, puis 3.7**.

Version 3.0

Après plus de 25 000 téléchargements, voici une nouvelle mise à jour majeure de cet ouvrage. Nous avons souhaité ajouter deux outils au panel déjà large des solutions : **Joomla** et **Drupal**, des outils assez répandus et avec des communautés actives et fertiles. Ces deux outils ont pour eux leur simplicité d'installation, de configuration et d'utilisation, ce qui a permis de créer une large base de références dans les mondes associatif, personnel et des petites entreprises.

Ces outils ont démontré leur stabilité et leur maturité. Leur simplicité, avec toutefois quelques fonctionnalités avancées, peut constituer un choix alternatif à des outils plus complets mais aussi plus complexe à mettre en œuvre.

En revanche, nous avons retiré les CMS **Cofax**, et **PHPNuke**, qui ne sont plus activement maintenus et dont les fonctionnalités sont aujourd'hui moins complètes que celles proposées par les autres solutions.

Les autres outils ont quant à eux profité de nombreuses améliorations depuis la précédente version de ce livre blanc. Quelques exemples :

- **eZ Publish 3.9** : support du clustering, gestion multilingue poussée, Single Sign-On ...
- **Typo3 4.1** : espaces de travail, contrôle d'accès front office, ...
- **InfoGlue 2.5** Editeur d'images, webservices, nouveaux modules d'authentification, ...
- **Jahia 5** : Back-office Ajax, nouveaux portlets, Import Export XML, workflows spécifiques multi-niveaux, compatibilité JSR168, métadonnées, éditeur FCKEditor, hibernate, cache ESI, clustering, ...

Version 4.0

20 000 téléchargements plus tard, nous vous proposons une nouvelle version, largement revue, de notre livre blanc sur les CMS open source.

Avant de parler des produits, nous avons souhaité mettre à jour les fondamentaux de la gestion de contenu. Les attentes ont évolué et de nombreux critères qui pouvaient être discriminants sont devenus standards dans la plupart des solutions CMS open source étudiées.

Les fondamentaux s'articulent désormais autour de 4 thématiques principales : la structuration des contenus, la manipulation des contenus, l'exploitation des contenus et la sécurisation des contenus. A ces sujets, nous ajoutons une analyse du socle technique qui est un aspect moins fonctionnel mais tout aussi important, surtout pour la réalisation de sites professionnels faisant face à de très fortes affluences.

Coté produit, on retrouve les mêmes têtes de liste : Drupal, eZ Publish, Infoglue, Jahia, Joomla, OpenCMS, Spip et Typo3. Tous ont globalement bien évolué avec un périmètre fonctionnel enrichi et un socle technique renforcé. Seule la solution Apache Lenya a été retirée de ce livre blanc. Bien qu'elle ne nécessite pas de base de données (fichiers XML) et qu'elle dispose d'une architecture intéressante, son interface utilisateur est trop pauvre et ne pourra convenir qu'à des utilisateurs d'un bon niveau technique.

Notons l'ajout d'une rubrique « Les CMS embarqués » présentant les briques de gestion de contenu incluses dans des solutions (portail, ged, ecm) qui ne sont pas des CMS. Elles pourraient être de véritables alternatives dans les années qui viennent.

Enfin, nous avons intégré une rubrique « Les prometteurs » qui, comme son nom l'indique, énumère des solutions brillantes en recherche de maturité qui ne devrait pas tarder à faire leur trou dans ce marché très fourni. Souvent, elles mettent en avant des attributs particulièrement différenciant : prise en main, montée en charge, architecture, etc.

Les CMS

Il est maintenant acquis qu'un site web, qu'il soit Intranet ou Internet, doit pouvoir être administré de manière interactive, c'est à dire qu'il doit offrir une interface permettant à ses responsables de définir de nouvelles pages, de nouvelles rubriques, de mettre à jour et réorganiser les informations.

Mais au delà des besoins propres à un site web, certaines entreprises identifient le besoin de constituer et d'administrer un *référentiel des contenus*, sur lequel s'appuieront différentes publications, tant sur le web que sur d'autres médias.

Naturellement, beaucoup d'outils peuvent répondre à ce besoin. Dans l'offre de produits CMS (Content Management Systems) on peut distinguer :

- d'une part les produits orientés vers la seule gestion d'un site web, simples, prêts à l'emploi, aux possibilités d'extension généralement limitées,
- d'autre part les produits plus haut de gamme, qui permettent de construire un véritable référentiel de contenus au niveau de l'entreprise, de définir les processus de gestion associés, et de décliner la restitution de ces contenus sur une variété de médias.

Aujourd'hui le premier domaine, celui des CMS intégrés, est totalement conquis par les produits issus de l'open source, tels que Joomla ou Spip. Ces produits sont d'une telle qualité et d'un tel dynamisme qu'aucune offre propriétaire ne peut rivaliser.

A l'inverse, le haut de gamme en matière de CMS était historiquement le domaine de produits commerciaux tels que Documentum, Vignette ou Interwoven. Mais aujourd'hui des solutions telles que Jahia, eZ publish, Typo3 ou Infoglupe, apportent des périmètres équivalents voire supérieurs. Elles sont devenues, au fil des années, plus que des alternatives sérieuses dans le domaine de l'*Enterprise Content Management*.

Depuis 2001 maintenant, Smile a bâti une expertise unique dans le domaine de la gestion de contenus, et en particulier une forte connaissance des meilleures solutions open source.

C'est la raison pour laquelle nous vous proposons ce livre blanc, qui présente un échantillon de cette expertise.

Soulignons que nous avons fait une sélection sévère en amont, de sorte que les produits évoqués ici constituent chacun une solution de qualité. Ce sont ensuite les spécificités de chaque projet qui orienteront vers l'une ou l'autre de ces solutions.

Les CMS open source

Un analyste du Gartner Group relevait qu'en matière de gestion de contenu, « *aujourd'hui les gens ne voient pas l'intérêt de payer les prix élevés des produits commerciaux les plus importants, et sont à la recherche d'alternatives de plus faible coût* ».

Les analystes s'accordent à penser que les avantages des solutions open source ne se limitent pas aux économies. La gestion de contenu, par sa nature même, requiert une part croissante d'adaptations. A travers le principe d'extension (plugins), les développeurs peuvent réaliser, en toute sécurité, des fonctions nouvelles telles que définir un nouveau type de contenus, un workflow, etc., des choses qui ne sont pas possibles avec des outils propriétaires.

Les CMS open source sont désormais adoptés par tous, notamment par les plus grands comptes, ainsi qu'en témoignent les références de Smile. Bien souvent, à l'occasion d'un projet de refonte, ou bien du renouvellement du contrat de support d'un produit propriétaire en place depuis quelques années, les alternatives open source sont considérées, et les avantages en particulier au plan économique, apparaissent de manière indiscutable.

Pérennité, support

Lorsque l'on choisit un produit sur lequel construire tout ou partie de son système d'information, la question de la pérennité est parfois plus importante encore que la liste des possibilités.

A cet égard, les solutions open source offrent deux avantages importants :

- Tout d'abord, leur survie ne dépend pas de leur rentabilité, ou de considérations marketing. Tant qu'une communauté s'y intéresse, le produit vivra.
- Ensuite, la libre disposition des sources, est un gage fondamental de pérennité, cela même si – il faut le souligner – il n'est nullement nécessaire que l'entreprise cliente les maîtrise.

Il faut garder à l'esprit que les solutions CMS propriétaires ont une dépendance exclusive avec leurs éditeurs. Ces derniers, peuvent fermer du jour au lendemain, en laissant leurs anciens clients démunis, n'ayant guère d'autre choix que de reconstruire en toute urgence leurs sites sur un autre outil. Les exemples semblables sont nombreux ces dernières années : un nom, un éditeur, une licence, même des références... ne sont pas des garanties de pérennité.

Certes le caractère open source n'est pas en soi une garantie d'éternelle jouvence, certaines solutions peuvent lentement perdre leur communauté et leur dynamisme, mais une chose est sûre : une solution open source ne coule *jamais* de cette manière, du jour au lendemain, et de plus une entreprise qui a construit sur une solution open source conserve toujours, avec les sources, de quoi faire vivre son système au moins quelques années.

Pour ce qui est du support, les spécificités des solutions open source sont maintenant bien connues. Elles offrent un support à deux voire trois niveaux.

D'un coté une communauté dont l'activité s'appuie sur un site de référence, où sont disponibles les téléchargements (downloads) et sources, bien sûr, mais aussi le plan d'évolutions (roadmap), les bugs répertoriés, les mailing-lists de support. Ce support communautaire est non contractuel, mais néanmoins très fort.

Ce support peut être complété du support de sociétés telles que Smile, qui lui peut être contractualisé.

Gestion de contenus – les solutions open source

A côté des solutions open source véritablement communautaires apparaissent de plus en plus des solutions qui sont le fait d'éditeurs, sociétés commerciales à but lucratif, qui ont fait le choix d'une diffusion sous licence open source. La libre diffusion permet à ces éditeurs de faire connaître leur solution le plus largement possible, et le business model s'appuie souvent sur une prestation de support et de conseil.

Ces sociétés proposent le plus souvent un support voire une garantie sur l'outil (prise en charge illimitée des anomalies sur le noyau).

SOMMAIRE

PRÉAMBULE.....	2
SMILE.....	2
CE LIVRE BLANC.....	3
QUELQUES RÉFÉRENCES DE SMILE.....	4
VERSIONS.....	5
Version 2.1.....	5
Version 3.0.....	5
Version 4.0.....	6
LES CMS.....	8
LES CMS OPEN SOURCE.....	9
PÉRENNITÉ, SUPPORT.....	10
SOMMAIRE.....	12
LES CMS OPEN SOURCE.....	14
NOTRE SÉLECTION.....	14
Drupal.....	14
eZ Publish.....	15
InfoGlue.....	17
Jahia.....	18
Joomla.....	19
OpenCms.....	20
Sip.....	21
Typo3.....	22
LES AUTRES.....	23
LES BRIQUES CMS.....	25
LES PROMETTEURS.....	26
CONCEPTS FONDAMENTAUX DE LA GESTION DE CONTENUS.....	27
STRUCTURER LE CONTENU.....	27
Structuration du contenu.....	27
Séparation du fond et de la forme.....	31
Multilinguisme.....	34
Organisation des contenus.....	36
Versions d'un contenu.....	39
MANIPULER LE	
CONTENU.....	42
Interface de gestion.....	42
Dépôt d'un contenu.....	44
Cycle de vie d'un contenu.....	48
Contribution collaborative.....	52
EXPLOITER	
LE CONTENU.....	54
Gabarits.....	54
Accessibilité.....	58
Référencement.....	60
Mesure d'audience.....	63
Syndication.....	65

Gestion de contenus – les solutions open source

<i>Personnalisation et animation de rendu.....</i>	<i>67</i>
<i>Recueil d'informations.....</i>	<i>69</i>
<i>Fonction de portail.....</i>	<i>70</i>
<i>Moteur de recherche.....</i>	<i>71</i>
DROITS ET SÉCURITÉ D'ACCÈS.....	75
<i>Utilisateurs, groupes et rôles.....</i>	<i>75</i>
<i>Annuaire.....</i>	<i>79</i>
<i>Identification unique.....</i>	<i>81</i>
SOCLE	
TECHNIQUE.....	83
<i>Framework.....</i>	<i>83</i>
<i>API distante.....</i>	<i>85</i>
<i>Plugins et extensibilité.....</i>	<i>86</i>
<i>Performance</i>	<i>88</i>
SYNTHÈSE.....	90

LES CMS OPEN SOURCE

Notre sélection

Cette rubrique présente notre sélection des meilleurs CMS du moment. Elle se base sur notre savoir-faire en matière de gestion de contenu, sur l'expérience de nos implémentations et sur les retours utilisateur de nos clients.

Bien sûr, d'autres solutions existent et certaines peuvent, d'ailleurs, mériter le détour ; n'hésitez pas à lire les rubriques suivantes qui présentent d'autres alternatives.

Les solutions ci-dessous respectent la plupart des exigences établies par l'équipe de veille de Smile – exemples : facilité de prise en main, socle technique robuste ou éprouvé, communauté dynamique, etc.

Drupal

Drupal est un CMS aux multiples facettes. Conçu à l'origine pour être un blog collectif, il trouve aujourd'hui des applications très variées : du portail communautaire au site corporate, en passant par l'intranet ou encore le site e-commerce.

La force de ce CMS est son extensibilité. Il accueille très facilement de nombreux modules complémentaires (forum, galerie photos, sondage, formulaire, newsletter, messagerie, chat, enquête, paiement en ligne, calendrier partagé, etc.). Il possède notamment des modules communautaires très soignés et appréciés. Les modules sont généralement faciles à développer grâce à une syntaxe accessible.

L'administration est une spécificité du produit ; **une seule et même interface est disponible pour la consultation (front office) et l'administration du site (back office)** ; un point qui plait particulièrement aux néophytes.

Au niveau de la gestion de contenus, Drupal se situe à mi-chemin entre Spip et eZ Publish. En effet, un module complémentaire (CCK) permet de **créer facilement des types de contenus structurés** (natif dans la version 7).

Au niveau de l'organisation des contenus, **Drupal a la particularité de travailler à partir de mots clés** (tags - taxonomie) et non autour d'une arborescence comme la plupart des CMS. Ce qui peut paraître un peu déstabilisant au début, apparaît comme une vraie force par la suite.

Drupal intègre un système de templates PHP très flexible, la gestion des droits utilisateurs, le multi positionnement des contenus, ainsi que des fonctionnalités de recherche et de statistiques.

Au niveau des performances, Drupal offre de très bons résultats notamment en mode déconnecté. Il constitue une véritable alternative aux CMS « hauts de gamme » notamment pour les sites purement éditoriaux.

La version étudiée ici est la 6.16.

Smile a déployé Drupal à plusieurs reprises dont pour NRJ-Studio, Diester, Alban Muller, BibNum (Cerimes), CDC Biodiversité, L'Equipe, GPSA, Bastide Diffusion, Autorisé de la Concurrence, ESTP, Université d'Orléans, Fondation EADS, INSA, etc.

eZ Publish

eZ Publish se distingue des autres CMS open source par ses possibilités de configuration et d'extension : il se présente comme un outil prêt à l'emploi mais aussi comme un framework de développement permettant de réaliser des applicatifs métier.

Sur les fondamentaux de la gestion de contenus, eZ Publish est l'un des outils les plus puissants disponibles en open source : contenus structurés, workflow, catégorisation, gestion des versions, etc.

eZ Publish permet de créer des types de contenus en ligne. En quelques clics et formulaires, il est possible de créer le type de contenu 'Compte-rendu de réunion' ou 'Actualité' ou encore 'Fiche bibliographique'. Le premier contenu sera composé d'un titre, du nom du projet associé, d'une date, d'une liste de participants, d'un corps de texte ... Le second contiendra uniquement un titre et un petit texte. Le troisième sera plus complexe.

eZ Publish pousse l'approche objet jusqu'au bout, en faisant de chaque contenu un objet et en lui appliquant toutes les méthodes disponibles : multi positionnement, versionning, multilinguisme, relation entre contenus, droits d'accès, workflow. **Cette notion en fait un CMS hautement paramétrable.**

Pour intégrer des contenus, le CMS eZ Publish propose de nombreuses alternatives : formulaire simple, multi-upload, accès webdav, frontend editing, chargement de documents Open Office.

Sur le plan technique, eZ Publish est en Php, fonctionne sur Apache et MySql (PostGreSql 8 et Oracle 11g sont supportés), gère trois caches différents en fichiers et sépare très distinctement contenus et présentation.

Sur le plan de la recherche, eZ Publish s'interface avec le moteur de recherche open source Solr ; **il en sort des résultats pertinents dans des temps de réponses très courts.**

La version étudiée ici est la 4.3.

Smile a déployé eZ publish sur plus de 100 projets, incluant Foncia, CFnews, Fia-Net, Afssaps, Evian Masters, Gallimard, BFM, Bouygues Telecom, Mazars, Yves Rocher, Groupama, Patrice Pichet, EMI, Voyages-Sncf, Eurostar.com, INRA, OSEO, Sport24, De Dietrich, Prisma Presse, Femme Actuelle, Elle, Gîtes de France, Hachette Filipacchi, Suez environnement, Amnesty International, National Geographic, Automobile Magazine, Forum des Images, Ordre des Pharmaciens, ...

InfoGlue

InfoGlue est un outil d'origine suédoise, qui dispose de nombreux atouts :

Des atouts technologiques : InfoGlue est développé en Java, et peut utiliser indifféremment les bases de données MySql, Oracle, Microsoft Sql Server ou DB2. **InfoGlue intègre l'outil de portail open source Pluto**, ce qui lui permet d'accepter des modules d'extensions. Concernant les workflows, **InfoGlue intègre l'outil open source OpenSymphony**, ce qui lui permet de configurer n'importe quel processus métier en Xml.

Des atouts fonctionnels : InfoGlue permet de définir sans développement de nouveaux types de contenus et de catégories, deux fonctions essentielles de la gestion de contenu haut de gamme.

La nouvelle version du CMS (3.0) est très attendue avec notamment une nouvelle interface graphique encore plus ergonomique.

En attendant, la dernière version (2.9.7.1) consolide les fonctions existantes et offre un périmètre déjà très riche : éditeur de texte, gestion multi-sites, publication de page à date donnée, versionning de page et d'arborescence, publication par lot, personnalisation de rendu, etc.

Bien que le projet ait débuté en 2003, il est encore peu connu en France ; on le trouve en particulier implémenté dans les universités.

Dans le monde Java, InfoGlue est une solution qui manquait à l'offre open source. Simple d'installation, simple d'utilisation, configurable et extensible, InfoGlue répond à de nombreux besoins, pour des sites web, mais aussi pour gérer des référentiels de contenus d'entreprise.

La version étudiée ici est la 2.9.7.1.

Smile a intégré la solution Infoglue à plusieurs reprises : Xinek, Oncoval, K par K, Lapeyre, Nouvelles frontières, Université de Pau, Université de Bordeaux 3 et 4, etc.

Jahia

Jahia est un produit franco-suisse, qui tient une place à part dans le monde des portails et des CMS JEE. Cela à deux égards.

En termes de licence tout d'abord. Il existe une version « Community », véritablement open source (GPL), mais qui n'est pas celle utilisée le plus souvent. Pour les développements professionnels, la Souscription Annuelle de Production est obligatoire. Elle permet d'obtenir : le droit d'usage annuel des versions Entreprise, la correction de bugs, le support de Production, etc.

En termes de positionnement ensuite, car **Jahia est une des rares solutions qui réunit portail JEE et gestion de contenu en un produit unique, parfaitement intégré.**

Jahia constitue une excellente alternative aux solutions de portail des grands éditeurs, la possibilité de voir et de modifier les sources de Jahia assurant la pérennité et l'adéquation de l'outil aux besoins.

Comme évoqué précédemment, la force de Jahia est de réunir CMS et portail en un tout parfaitement packagé et facile à administrer. Le respect des standards techniques, notamment de la norme JSR 168 (et 286), assure une réelle compatibilité et capacité à intégrer des « portlets » tierces ou spécifiques, donnant accès aux applications internes de l'entreprise.

Jahia est **prêt à l'emploi** et ne nécessite le plus souvent aucun développement dans sa mise en œuvre (hormis son intégration). Librement téléchargeable, Jahia est d'un niveau de finition et de packaging impressionnant : il s'installe en quelques minutes, prêt à fonctionner.

Jahia se distingue des autres CMS par son interface d'administration des contenus, qui est fondue dans le site lui-même. Une fois identifié, vous avez accès aux fonctions d'ajout, modification et suppression, directement sur le site, à l'intérieur des pages et des menus. Ce mode de gestion apporte une réelle simplification pour le contributeur, qui voit et comprend directement où il doit modifier l'information.

Au niveau fonctionnel, Jahia est une des solutions CMS les plus abouties : gestion multi-sites, versionning, workflow efficaces, données structurées, multilinguisme, gestion des droits très fine, etc. La version 6 a notamment apportée une interface utilisateur très agréable construite via l'Ajex notamment.

La version étudiée ici est la 6.

Smile a mis en place le portail Jahia pour Bouygues Immobilier, BNP Paribas, Valorissimo, Arjowiggins, Beauté Prestige International, Veolia Eau, Eutelsat, Egide, Académie de Rennes, Conseil Régional d'Ile de France, SANEF, Groupe COLAS, Adoma, Fondation d'Auteuil, Académie de Créteil, IRP Auto, Pimkie-Diramode, AFNOR, OCP, Vinci Energies, Conseil Général de la Côte d'Or, Diagnostica Stago, Conseil Régional du Centre, Conseil Supérieur du Notariat, Macif, La Poste...

Joomla

Joomla est un CMS développé à partir de Mambo. Il a été créé suite à un différent entre les développeurs principaux et la société coordinatrice des développements. Aujourd'hui, la majorité des développeurs de la communauté se consacrent à **Joomla**, ce qui fait nettement pencher la balance en sa faveur.

Cet outil se démarque principalement par la **convivialité de son interface d'administration**. Le mot d'ordre étant de « donner un contrôle total du produit à un non-technicien ». Création de pages, catégorisation, recherche, statistiques d'accès, urls significatives ainsi que de nombreux modules sont directement intégrés et ne demandent pas de connaissances spécifiques pour leur mise en œuvre.

Cependant, Joomla ne permet pas de gérer nativement différents types de contenus, et ne pourra pas être utilisé pour un besoin fort de structuration des contenus (même si il existe un plugin permettant la mise en œuvre des contenus structurés).

Ce CMS conviendra parfaitement pour des sites personnels mais pourra également répondre aux besoins de certains sites professionnels ; ceux notamment de type Corporate (simple publication).

La version étudiée ici est la 1.5.15.

OpenCms

OpenCms est un CMS open source édité par la société allemande Alkacon. C'est un produit d'envergure, solide, qui représente un important effort de développement.

La dernière version, la 7.5, offre de nombreuses fonctionnalités : contenus structurés, publication par lot, publication à date donnée, éditeur de contenu, gestion des liens, webdav, délégation de droits, etc.

OpenCms est construit sur un framework Java spécifique, dans une perspective d'extensibilité. Ainsi, même si les fonctions de base sont prêtes à l'emploi, les mises en œuvre sophistiquées induiront du développement sur le principe de l'extension des classes Java de base.

L'une des caractéristiques de l'outil est un back-office dont **le graphisme reproduit pratiquement une interface Windows** : arborescence, menus contextuels, boîtes de dialogues, etc. Très séduisante sur le plan esthétique, l'interface est un peu lourde à l'usage. Depuis la version 6, l'outil permet aussi d'éditer les contenus en front-office, sans passer par le back-office. Mais cet usage reste réservé à des cas très simple, et non pour l'administration complète des contenus.

OpenCms possède de belles références de sites 'corporate', dont une majorité en Allemagne.

Citons notamment le site Alice DSL : www.alice-dsl.de, le site de Bonduelle Germany : www.bonduelle.de, le site de Fujifilm France : www.fujifilmgraphic.fr, le site de Intersport International : www.intersport.com ...

La version étudiée ici est la 7.5.2.

Smile a mis en œuvre OpenCms pour l'UDF et pour Editions Francis Lefebvre.

Spip

Spip dispose des principales fonctionnalités que l'on peut attendre d'un CMS. A l'origine, Spip était principalement utilisé par des internautes pour gérer leur site web personnel, ou pour des sites d'associations, mais Spip a grandi et est maintenant utilisé par des organismes privés ou publics, pour gérer leur site web professionnel. Spip est un projet open source français, et sa communauté peut être qualifiée d'active, avec plusieurs versions par an et des centaines – voire des milliers – de membres.

Spip fait partie des rares CMS à pouvoir prétendre à **plusieurs milliers de références** à travers le monde. Cette réussite s'explique par **la simplicité du produit** : simplicité d'utilisation, mais aussi simplicité de déploiement et d'adaptation.

En contrepartie de cette simplicité, Spip présente quelques limitations sur des fonctionnalités clés de gestion de contenus, ce qui le limite clairement au monde des outils de gestion de contenus orienté 'web', et non pas 'entreprise'. Notons par ailleurs que Spip dispose d'une interface d'administration entièrement multilingue, traduite dans plus de 20 langues, comprenant l'arabe, l'esperanto, le russe et même le chinois.

En parallèle à Spip, s'est développé le projet **Spip-Agora**, sponsorisé par un service du gouvernement français, avec la volonté d'ajouter des fonctionnalités au projet Spip pour l'utiliser au sein de l'administration pour des projets plus complexes. Après avoir tiré le produit vers haut, le projet Spip-Agora a cessé et n'est plus supporté désormais (depuis mai 2008).

Près de 2 ans après la sortie de la version 1.9.2, Spip a sorti sa version 2.0 ; un nouveau souffle. Cette nouvelle branche offre de nombreuses nouveautés : installation possible sur PostGreSQL et SQLite en plus de MySQL, interface privée simplifiée basée sur l'Ajaxy, page de téléchargement et d'installation de plugins, gestion des conflits, API et nouvelles fonctions pour le développement de templates.

La version étudiée ici est la 2.0.10.

Smile a construit plusieurs sites avec Spip, notamment un générateur de sites Spip pour les délégations de la Croix-Rouge Française, le site du Fratel, réseau francophone de la régulation des télécommunications, ANCV, Evian masters experience, ou encore plus récemment le Secours Catholique.

Typo3

Typo3 est le fruit de plusieurs années de travail d'un gourou danois du nom de Kasper Skårhøj. Le produit est sorti fin 2000, et depuis une communauté très active s'est développée, particulièrement dans les pays germaniques.

En termes de fonctionnalités prêtes à l'emploi, Typo3 est l'un des outils les plus riches que nous ayons trouvés à ce jour. Il offre à peu près tout ce que l'on peut souhaiter, et cela avec un bon niveau de finition. Gestion des droits et des contributions, cache, habilitations, gabarits, etc., tout y est, avec peu de limitations.

Parmi les fonctionnalités offertes par Typo3, on peut citer la manipulation d'images, qui permet de redimensionner des images, de créer des vignettes et également de générer dynamiquement des titres en tant qu'images.

L'une des grandes forces de Typo3 réside dans son **extensibilité par modules**. Un module peut ajouter un ensemble de fonctionnalités à Typo3, mais aussi modifier une fonctionnalité déjà intégrée, sans modifier le code de Typo3, et donc, le laissant compatible avec les futures versions de Typo3.

Plusieurs milliers de références opérationnelles sont indiquées sur le site de la communauté. En France, Smile a déjà mis en place plus de 100 sites sous Typo3, et assuré un grand nombre de formations et transferts de compétences pour aider d'autres intégrateurs.

Typo3 a notamment été choisi par Dassault Systèmes pour construire son Intranet. A cette occasion Smile, a eu le privilège de travailler aux cotés de Kasper sur ce projet ambitieux et passionnant.

La version étudiée ici est la 4.3.

Smile a mis en œuvre Typo3 pour la CNIL, Dassault Systèmes, Eco-Emballages, Areva, AMUE, Archimag, BHV, Armée de l'air, IPSOS, CCI, Polytech'Savoie, PCI, Projectif, Spie, Faceo, Institut National de l'Audiovisuel, Cidil, RATP, Souriau, Prolea, PSA Peugeot Citroën, Crédit Coopératif, Château de Versailles, SAS, ETS Europe, Agence de l'Eau Seine Normandie, Glamour, Jaccede.com, IUFM Créteil, Conseil Régional d'Ile de France, ODIT France, AFPA, Action contre la Faim, Arvalis, Comexpo, CORA, MGEN, World Water Council, Société Générale, Immovalor, Legris Industries, Agritel, My Coop, Gefco, Salon de l'Agriculture, INVIVO, Cidilait...

Les autres

Il existe de nombreux autres CMS au-delà de ceux que nous avons sélectionnés ; parmi eux, on trouve Mambo, Apache Lenya ou encore Zope, des CMS que nous avons retenus les années précédentes. Nous faisons ici un rapide rappel de leurs offres.

Mambo est un outil de gestion de sites web relativement complet. Il est « l'ancêtre » de Joomla. Comme les autres outils de ce type, ils disposent de nombreux modules prêts à l'emploi, et pratiques pour des sites de communautés : livre d'or, forum, galerie photo, blog, etc. Sa simplicité d'accès permet de créer des sites graphiquement réussis, même pour des non-informaticiens.

La dernière version de Mambo, la 4.6.4, date de juillet 2008. Le projet semble gelé ; la plupart des contributeurs s'étant orientés vers Joomla.

Apache Lenya est un outil à part dans le monde de la gestion de contenus. Intégré à la fondation Apache, il est issu du produit Wyona CMS, d'origine suisse.

Lenya est un logiciel de gestion de contenus ou de documents, développé en Java, basé sur le framework Cocoon et manipulant des contenus Xml.

L'une des particularités de Lenya est que toutes les données qu'il gère sont stockées dans des fichiers Xml. Même les informations de structure comme les groupes et utilisateurs sont gérées en fichiers Xml. Lenya ne nécessite pas de base de données pour fonctionner. Les fichiers Xml sont organisés dans une arborescence et sont affichés en Html - ou dans un autre format - au moyen de feuilles de styles Xsl. La modification des contenus Xml (dont les schémas sont modifiables) se fait par des éditeurs graphiques riches intégrés à l'interface web, issus de projets open source séparés.

Son architecture particulière peut être utile dans le cadre d'un besoin spécifique (pas de base de données par exemple) mais son offre fonctionnelle couplée à sa difficulté d'utilisation repousseront plus d'un contributeur face à l'offre de qualité de ses concurrents.

Enfin, on ne peut parler de CMS sans mentionner **Zope**, à moins de s'attirer les foudres des nombreux *aficionados* de ce produit.

Zope est un serveur d'application très complet, qui va bien au delà de la fonction de CMS. Tout le monde s'accorde à penser que d'être construit en environnement Python est pratiquement son seul défaut. Il est vrai que les compétences sont rares pour faire du développement ou de la maintenance en Python. C'est la raison pour laquelle nous avons choisi de ne pas retenir Zope parmi les solutions CMS privilégiées, car nous voulons être en mesure de compléter ou d'adapter les fonctions natives de l'outil lorsque cela est nécessaire. D'ailleurs l'arrêt de Zope CPS par son principal promoteur Nuxeo a confirmé l'abandon progressif de cet environnement et ce malgré la sortie d'une nouvelle version.

Les briques CMS

Au delà des CMS « métiers », **il existe un nombre important de solutions Web qui mettent en œuvre une « brique » CMS** (gestion de contenu). Cette brique n'est alors qu'un module parmi l'éventail de ceux proposés.

C'est le cas des portails, comme Liferay ou ExoPlatform, qui au-delà de leurs fonctions premières d'agrégateur de contenu proposent également des fonctions de gestion de contenu (CMS). **Ces briques ne sont pas encore aussi évoluées que les meilleures des CMS mais elles évoluent rapidement** et on assistera peut-être dans quelques années à un rapprochement général des solutions de type CMS et de type Portail, à la manière de Jahia.

A l'inverse, d'autres solutions ne sont pas conçues pour la mise en œuvre de sites Web complexes, elles n'incluent pas, de base, de réelles briques CMS. C'est le cas de Wordpress qui est un des leaders des solutions de blogs. Ce type d'outil serait tout à fait capable, dans un futur proche, de concurrencer les meilleurs des CMS ; surtout avec la force de frappe qu'on lui connaît. Il a déjà pour lui des notions de structuration de contenu et l'expérience de la montée en charge.

Enfin, plus étonnant encore, on s'aperçoit que **certains outils de GED ou plus exactement d'ECM seraient capables d'élargir leur périmètre pour fournir des fonctions de publication avancée**. On pense notamment à Nuxeo qui est capable de modéliser des articles ou des actualités (des contenus plus largement). Ces outils sont très puissants dans la structuration (création de type de document complexe) et la manipulation de données (multitudes de chargement possible dont des interfaçages très réussis avec Microsoft Office ou Open Office). Seule la partie Publication des contenus n'est pas ou peu développée. On citera le Webengine de Nuxeo qui permet de publier des mini-sites Web, conçus comme des dossiers dans l'arborescence des contenus. Et le module WCM d'Alfresco qui gère la production des contenus et les pousse vers un CMS pour la publication.

Notons enfin que la norme CMIS est en passe de permettre un pont simple et efficace entre des contenus gérés dans un ECM et à publier dans un CMS.

Comme on le voit, la frontière de ces outils est de plus en plus mince ; et, il ne serait pas étonnant de voir une nouvelle distribution des rôles dans les années à venir.

Les prometteurs

Smile s'engage à servir le meilleur des produits open source à ses clients. Pour cela, nous réalisons une veille constante des outils de gestion de contenu. Certains tendent à s'effacer, d'autres au contraire sont particulièrement dynamiques dont quelques nouvelles solutions.

La plupart du temps, elles sont trop jeunes pour être employées dans le cadre d'une réalisation professionnelle mais nous gardons un œil sur elles et nous espérons leurs développements.

On peut citer :

- SilverStripe, Modx, CMS Made Simple, pour la qualité de leur prise en main notamment
- Sympal, Diem, Apostrophe, jeunes produits basés sur le framework Symfony
- Cahaya, produit basé sur Zend Framework
- Magnolia, produit qui tarde à trouver sa communauté
- Silverpeas, produit passé récemment au modèle Open Source

Les CMS basés sur un framework ne sont pas particulièrement avancés à ce jour mais offrent de bonnes perspectives. Ils proposent surtout des alternatives prometteuses notamment pour les projets nécessitant des modifications profondes des CMS existants.

CONCEPTS FONDAMENTAUX DE LA GESTION DE CONTENUS

Structurer le contenu

Dans nos précédentes versions de ce livre blanc, nous traitions d'un bloc les aspects de structuration et de manipulation du contenu. Les deux points sont très liés c'est indéniable ; mais la notion de structuration de contenu a pris une telle place dans le choix d'une solution CMS qu'il nous semblait important d'en détailler les concepts fondamentaux.

Structuration du contenu

Un article est un élément de contenu. Selon les cas, il peut représenter entre quelques lignes et quelques pages de texte, traitant d'un sujet donné.

Un article n'est pourtant pas la plus petite unité de structuration du contenu : un article est lui-même généralement décomposé en champs élémentaires, qui peuvent être par exemple : *titre, sous-titre, image, corps de texte*.

On peut distinguer deux manières de structurer ainsi les articles : une structuration fonctionnelle et une structuration sémantique.

La structuration fonctionnelle va décomposer l'article sur la base du rôle, de la *fonction*, assurée par chaque item, typiquement *titre, sous-titre, résumé*. Cette fonction est indépendante du sujet traité, du thème de l'article : que l'on parle de recette de cuisine ou de match de foot, on peut identifier un champ titre et un champ résumé.

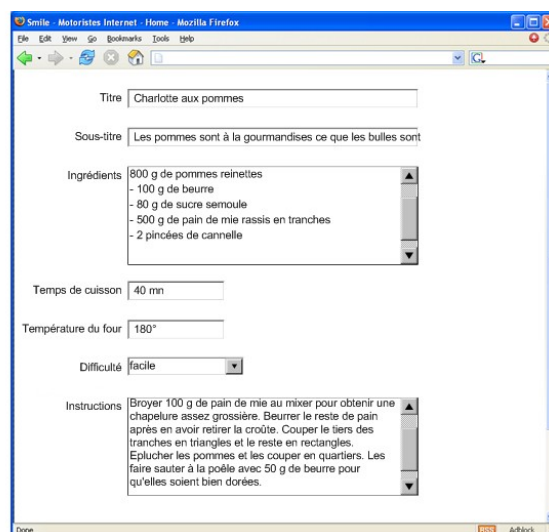
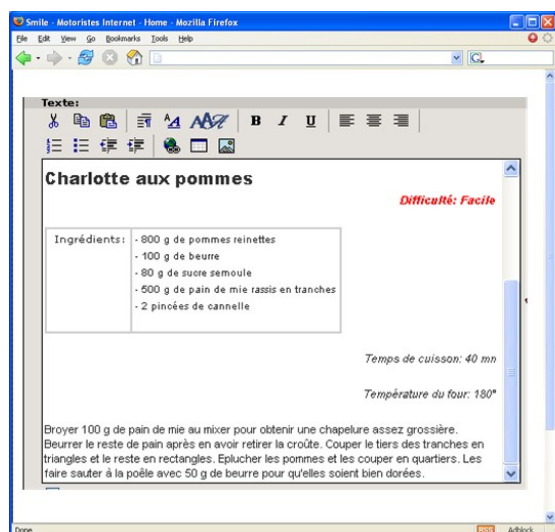
La structuration sémantique décompose l'article sur la base de la signification de chaque item. Pour une recette de cuisine, on pourra distinguer des champs *ingrédients, temps de cuisson, température du four, niveau de difficulté*, etc. Pour des matches de foot : *équipes, stade, date et heure*, etc. Comme on le voit, la structuration sémantique dépend étroitement du thème de l'article, de sorte qu'au

sein d'une même gestion de contenu, on peut être amené à définir plusieurs structures ou *types d'articles*. Chaque contenu étant d'un type ou d'un autre, il ne sera pas toujours simple de restituer un article d'un type A là où un type B était attendu.

Le fait de bien structurer chaque article de manière spécifique présente des avantages déterminants :

- Cela permet de bien **guider la saisie** des articles au moyen d'un formulaire adapté, et donc de s'assurer qu'ils entrent tous dans le moule correspondant à leur type. C'est donc un facteur de qualité des contenus.
- Cela permet également de distinguer, lors de la publication, chacun des composants de l'article, et donc de **véritablement séparer la forme du fond**, telle que nous l'avons évoqué plus haut. Il sera possible de définir dans la maquette de restitution de l'article (gabarit), que le champ ingrédients est à présenter en italique, tandis que le temps de cuisson est en gras. Et l'on pourra effectivement modifier cette restitution sans toucher aux contenus des recettes de cuisine. En l'absence de structure d'article adaptée, il aurait fallu soit renoncer à ces effets sur les ingrédients et le temps de cuisson, soit injecter de la mise en forme dans le corps de l'article.
- Cela permet d'effectuer des recherches précises au-delà de simples recherches textuelles ; on pourrait ainsi récupérer toutes les recettes avec un temps de cuisson inférieur à 30 minutes.
- Enfin, c'est ce qui **donne du sens aux contenus**. Les contenus ne sont plus des champs de texte, ils sont des attributs qualifiant un objet : pays, ville, prix, date, etc... et ce sont donc toutes les possibilités d'utilisation de ces contenus qui sont décuplées.

On voit donc bien l'intérêt qu'il y a dans cette manière de gérer les structures d'articles. Mais il faut bien mesurer que ces avantages sont obtenus au prix d'**une certaine complexité**. Le CMS devient alors une véritable base de données : on définit de nouvelles entités, les attributs de ces entités, les contrôles à effectuer sur chacun des champs, les relations entre ces entités, etc... Lorsque la déclaration d'un type de contenu structuré peut se faire sans développement informatique, alors **c'est une des fonctionnalités les plus appréciables d'un CMS**.



Sur les images précédentes, à gauche la « Charlotte aux pommes » version non-structurée du contenu, et à droite la même en version structurée. On conçoit aisément que la maîtrise de la restitution comme de l'utilisation des contenus sont beaucoup plus grandes en présence de contenus structurés.

Ce concept de la gestion de contenu (structuré ou non structuré) est donc capital dans le choix de votre solution CMS. Il est important de bien réfléchir à ce point et de vérifier quelle type d'informations vous souhaitez publier et que ce vous souhaitez en faire.

Nativement **Joomla** et **Spip** ne permettent pas de définir des types d'articles spécifiques (recette de cuisine, etc). Il s'ensuit quelques limitations, mais aussi une grande simplicité dans la gestion et l'organisation d'un site. Tous les contenus sont saisis sur un même formulaire, tous les contenus peuvent être restitués sur un même gabarit. Toutefois, il est à noter que des extensions de qualité diverse existe pour « contourner » ce problème : jSeblod CCK pour Joomla (relativement complet) et Champs Extras 2 pour Spip (qui permet d'étendre les attributs des types de contenu existant uniquement).

Drupal supporte nativement les contenus de types différents, mais ne propose pas de gérer directement leur édition. Un module optionnel (CCK) rajoute cette possibilité et permet de créer simplement de nouveaux types de contenu à l'aide de formulaires intégrés au back office. Le succès de cette extension fait qu'elle est désormais incorporée nativement dans la version 7 de Drupal.

Le module TemplaVoila de **Typo3** permet de définir de nouvelles structures de contenus, avec des formulaires de saisie adaptés. Si la création reste un peu technique, le résultat est bien là, on peut créer de nouveaux types de contenus, et les utiliser simplement partout dans le site. Les contenus structurés sont stockés en XML dans un nouveau champ de la table des contenus. Il est particulièrement adapté aux sites nécessitant un fort besoin de structuration de contenu.

Depuis la version 6, **OpenCms** permet de définir des types d'articles spécifiques, basés sur des schémas XSD. Les structures possibles restent simples et n'utilisent pas tout le potentiel du XML, mais il est possible de créer simplement de nouvelles structures de contenus. Les contenus structurés sont ensuite stockés en XML, dans un champ de la base de données.

eZ Publish se distingue dans ce panel. C'est l'outil qui permet le plus aisément de définir, en ligne, tous les types de contenus que l'on souhaite, simplement via l'interface web. Par défaut, les nouveaux types de contenus ont un gabarit standard qui affiche tous les champs du contenu. Pour chaque type de contenu, il faut ensuite définir un gabarit de prévisualisation pour l'interface d'administration et un autre pour l'interface de restitution (front office). Notons que eZ Publish permet également de définir des relations entre les contenus.

Jahia permet de définir des nouveaux types de contenus par le biais de fichiers CND dans lesquels on définit la structure du contenu. Il n'y a pas d'interface graphique pour créer des nouveaux types de contenus, mais la création reste simple et peut se faire en même temps que la définition des nouveaux gabarits de rendu. Jahia 6 a apporté une amélioration sur ce point.

InfoGlue permet de créer de nouvelles structures de contenus via son interface, sans nécessité de connaissance technique. Et InfoGlue va même plus loin, en étant le seul CMS permettant de définir des règles de validité pour chaque champ, en utilisant les expressions régulières. Ceci va permettre, par exemple, d'imposer que la première lettre soit en majuscule, ou qu'il n'y ait pas d'espace, ou encore que le champ soit un entier suivi d'une lettre ...

Séparation du fond et de la forme

L'un des principes des CMS est d'instituer une frontière entre contenant et contenu, entre la forme et le fond, entre la présentation graphique des articles, et le texte des articles (ou des attributs de ces articles).

Il y a quantité de bonnes raisons de préserver cette séparation.

D'abord forme et fond relèvent de **métiers différents** : la mise en page est élaborée par un graphiste, tandis que le contenu est fourni par un contributeur.

Ensuite, l'un et l'autre doivent pouvoir **évoluer indépendamment**. Il doit être possible en particulier de changer le design d'un site sans avoir à intervenir sur les contenus. Et réciproquement, d'ajouter ou de modifier un texte sans se préoccuper de sa mise en forme.

Enfin une mise en page unique (gabarit) est partagée par de multiples contenus: c'est ce qui fera **l'homogénéité du site**. Elle serait impossible à obtenir si chaque saisie de contenu devait définir également la mise en page.

Ajoutons aussi que la mise en forme se définit toujours en référence à **un média de publication**, de sorte que mêler mise en forme et contenus implique de réduire les possibilités de publication à un seul média.

La chose est entendue donc : d'un coté la forme, de l'autre le fond. De manière pratique, cela signifie que les textes insérés dans la base de contenus ne doivent pas comporter de mise en forme.

Cette contrainte est parfois trop forte pour certains usages, et il faut alors autoriser quelques exceptions : gras, italique ; soit le niveau de mise en forme accepté au sein d'un article de journal. Mais pas plus : fonte, taille de caractères, couleurs, alignement, agencement en colonnes, ... tout cela relève de la maquette et ne doit pas être autorisé au sein du contenu. **A noter, que certains éditeurs WYSIWYG (voir plus loin) stockent l'ensemble des contenus (même les gras, ...) sous format XML et proposent donc une séparation complète du fond et de la forme.**

Pour répondre au besoin de mise en forme, les CMS incluent en général un outil d'édition graphique (WYSIWYG), une sorte de « mini-word », qui permet de mettre en forme les textes, mais aussi d'insérer des liens et des images. L'intégration d'un tel outil peut être plus ou moins complète, selon qu'il sache prendre en compte les styles CSS définis pour le site, que l'insertion de lien permette de sélectionner un des contenus du site ou que l'insertion d'image permette d'insérer une image de la médiathèque.

Pour résumer, la séparation du fond et de la forme est importante pour réaliser de la publication multi canal (web et mobile par exemple). De plus, elle est un impératif à la réalisation de sites accessibles.

Joomla propose nativement la solution open source TinyMCE (Project of the Month, Sourceforge, Janvier 2009), éditeur wysiwyg de bon niveau qui produit du code Html de qualité. D'autres éditeurs peuvent être mis en place à travers le principe d'extensions.

Drupal n'intègre pas nativement d'éditeur texte riche, mais des modules complémentaires permettent de rajouter facilement cette fonctionnalité en utilisant les éditeurs open source les plus répandus. L'intégration de l'éditeur externe est alors simple, par exemple sans gestion des liens vers les contenus.

OpenCms dispose nativement d'une solution d'édition graphique (FckEditor), bien intégrée à l'outil.

Il y a quelques versions de cela, **Typo3** avait opté pour une solution VBScript qui ne fonctionnait que sous Internet Explorer. Dans sa version 4, Typo3 intègre maintenant l'éditeur wysiwyg HTMLArea RTE, ouvrant ainsi la compatibilité à tous navigateurs. Cet éditeur est parfaitement intégré, avec gestion des images et personnalisation possible de l'éditeur selon les utilisateurs.

Concernant **Spip**, il est à remarquer qu'il n'y a pas d'éditeur wysiwyg et qu'il est remplacé par des balises spécifiques que l'utilisateur doit connaître (un « mini-éditeur » permet de les insérer). Ceci limite beaucoup les possibilités d'édition des pages, comme l'inclusion des images dans le texte, un tableau ou une structure plus complexe ; cela permet cependant une complète séparation entre le fond et la forme. Notons toutefois qu'il est possible d'intégrer un éditeur wysiwyg complet tel que FckEditor.

Jahia intègre l'éditeur graphique FckEditor, ce qui assure un fonctionnement sur toutes les plateformes client. A noter qu'il existe trois niveaux définissables d'utilisation de cet éditeur (légère, complète, etc.) configurables profil par profil. On peut ainsi proposer, par exemple, la version complète à l'administrateur et la version simplifiée aux contributeurs.

eZ Publish intègre son propre éditeur "Online editor", reversé en open source sous licence GPL. Cet éditeur wysiwyg écrit en fait du XML au lieu de Html, ce qui impose de très légères limitations dans la mise en forme, mais assure une parfaite séparation de la forme et du fond. De plus, parfaitement intégré au CMS, cet éditeur permet d'insérer des fichiers issus de la médiathèque eZ Publish, d'uploader un fichier directement en fichier joint du contenu. La dernière version, la 5.0 de cet éditeur, est basée sur TinyMCE.

InfoGlue intègre nativement un éditeur texte riche, basé sur FckEditor (les versions antérieures à la version 2 utilisaient HTMLArea), très bien intégré au CMS, et qui permet de créer un lien vers un contenu en le sélectionnant, de même pour l'insertion des images.

La bonne séparation de la forme et du fond est en fait davantage une question de politique qu'une question d'outils.

Sur un CMS intégré tel que **Spip, Joomla, OpenCMS voire Drupal**, il n'est pas déraisonnable d'autoriser les contributeurs à définir des mises en forme au delà du gras et de l'italique. La contrainte du multi-canal est moindre, et l'on peut compter sur le processus de validation pour contrôler ces contributions et assurer l'homogénéité des pages.

A l'inverse, dans une optique de CMS d'entreprise, tels que **eZ Publish, InfoGlue, Jahia et Typo3**, cette contrainte devrait être respectée avec rigueur. Il faut souligner que les possibilités de structuration de types d'articles spécifiques limitent sensiblement le besoin d'insertion de mise en forme dans les articles, puisque pour chaque champ d'article une mise en forme pourra être définie au niveau des *gabarits*, comme on le verra plus loin.

Multilinguisme

Pendant très longtemps, le multilinguisme était mis en œuvre artisanalement à partir d'astuces bâties, entre autres, sur l'arborescence du CMS. On devait ainsi créer autant de rubriques (exemple : Fr et En) à la racine du site que de langues pour lui donner une dimension internationale. Les sites correspondant vivaient alors de façon autonome sans aucune contrainte ni liaison.

Pour les besoins de nos entreprises et organisations présentes à l'international, une meilleure prise en charge du multilinguisme est nécessaire.

Ainsi, pour construire des sites multilingues, le CMS doit gérer, pour chaque contenu, différentes traductions - on retrouve là une notion de liaison/de dépendance. Sans cette notion, impossible de basculer d'une langue à une autre sur une page donnée, impossible d'utiliser un outil de traduction automatique des pages, etc... ; il manque alors une liaison entre les contenus traduits.

La gestion du multilinguisme peut devenir complexe : champs à traduire et champs sans traduction, circuits spécifiques pour la validation des traductions, traduction automatique ou semi-automatique, droits spécifiques pour une langue, prise en compte de différents « alphabets », reconnaissance de la langue du visiteur, etc.

C'est pourquoi, il s'agit désormais d'une fonctionnalité fondamentale de la gestion de contenu. Aucun outil n'offre la même implémentation du multilinguisme ; certains se limitent au même conteneur pour les différentes langues ; d'autres proposent une liaison complète avec des fonctionnalités efficaces. Soyez particulièrement attentif à ce point si vous souhaitez communiquer à l'international.

Pour mettre en œuvre un site multilingue sous **Drupal**, il faut activer un plugin (content translation et locale). A partir de là, vous pouvez configurer les langues et les comportements de votre site. Comme **SPIP**, **Drupal** effectue des liens entre les différentes versions linguistiques, un principe de base mais généralement suffisant.

eZ Publish gère parfaitement le multilinguisme ; c'est un des plus puissants dans ce domaine. Chaque contenu peut être traduit en une ou plusieurs langues. La création de versions linguistiques d'un site est très facile à mettre en œuvre. Des règles de gestion configurables permettent de définir le comportement d'eZ Publish en l'absence d'une version linguistique d'un contenu.

Infoglu propose une gestion des langues très plaisante. Les interfaces sont claires et les versions linguistiques se font autour d'un contenu et non d'une page. Un contenu ou une structure (page) est composée de 2 facettes : couverture et version. La couverture contient les informations générales du contenu (est-ce que le contenu est protégé ? quel est son cycle de vie ?, ...). Chaque contenu et structure possède une et une seule couverture. Chaque élément possède de multiples versions. Une version est créée pour un élément dès que l'on modifie un contenu publié ou lorsqu'on crée une nouvelle déclinaison de langue. Pour chaque élément, on aura ainsi une arborescence de version, une branche par langue. Ce CMS est très complet sur ce point.

Jahia gère parfaitement le multilinguisme. Le mécanisme linguistique est bien rodé et facilement accessible au travers des différents écrans. Il gère les langues « par défaut » en fonction des préférences utilisateurs (lorsque le profil est connecté par exemple). L'administrateur peut configurer les règles d'affichage en front office selon la disponibilité linguistique des contenus. Jahia permet de déclarer des attributs de contenu comme non traduisible. Une gestion multilingue très complète.

Joomla ne gère pas nativement le multilinguisme de ses sites. Pour mettre en place un site en plusieurs langues correctement (au-delà de deux arborescences distinctes), il faut activer une extension complémentaire comme Joom !Fish.

OpenCMS gère également le multilinguisme. A la manière de Jahia, il peut exister différentes versions linguistiques pour un contenu donné.

Spip gère le multilinguisme, il a même bien évolué depuis ses précédentes versions. Spip étant orienté « page » davantage que « contenu », la traduction d'un article se fait par lien entre deux pages (renseigné par le contributeur) ; un mode simple mais suffisant dans de nombreux cas.

Typo3 propose plusieurs alternatives pour mettre en place un site multilingue. Outre celle consistante à développer X sites (un par langue), il est possible de définir plusieurs versions linguistiques pour une page donnée (une page possède alors des contenus en français et en anglais par exemple). Pour l'initialisation d'une version linguistique, le contributeur peut copier le contenu depuis une source existante. A partir de là, l'administrateur du site peut déléguer l'administration d'une version linguistique à un contributeur donné.

Organisation des contenus

Le mode d'organisation le plus classique est bien sûr celui d'une structure hiérarchique arborescente, semblable à la structure de répertoires d'une arborescence de fichiers. Chaque niveau d'arborescence, chaque division en répertoire, correspond à une logique, un classement par rapport au sens de chaque item, par exemple en référence à des thèmes.

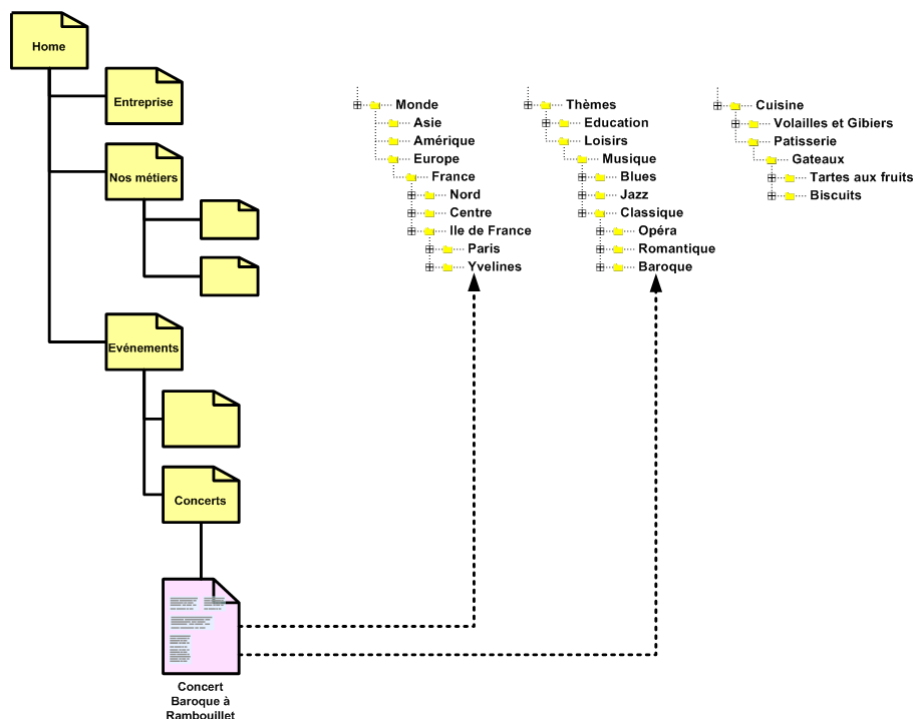
Ainsi par exemple on pourra organiser le contenu d'un site de média en *sport, culture, international, ...*, puis diviser la rubrique *sport* en *basket, football, ...*

Dans cet exemple, l'organisation des contenus est en correspondance avec l'organisation de la publication. Ce n'est pas une obligation, mais on comprend bien que cette correspondance entre organisation interne et organisation externe simplifie considérablement le travail du gestionnaire.

Pourtant lorsque la gestion de contenus s'envisage au niveau de l'entreprise comme socle de plusieurs publications, il est peu probable que l'organisation des publications soit à l'image de l'organisation du référentiel de contenus. Dans ce cas, il convient de définir de quelle manière les articles viennent se placer dans la structure de chaque publication, soit **de manière explicite**, en positionnant chaque article, soit **au moyen de règles de gestion**.

Même au sein d'un unique site, **l'organisation hiérarchique pure n'est pas toujours satisfaisante**, et peut être trop structurante. Il y a des alternatives. On peut par exemple associer un article à différents thèmes, ce qui crée une organisation plus relationnelle, ou ensembliste, que hiérarchique. C'est ce que l'on appelle parfois la **catégorisation** des contenus ou taxonomie, qui est une fonctionnalité importante de la gestion de contenus.

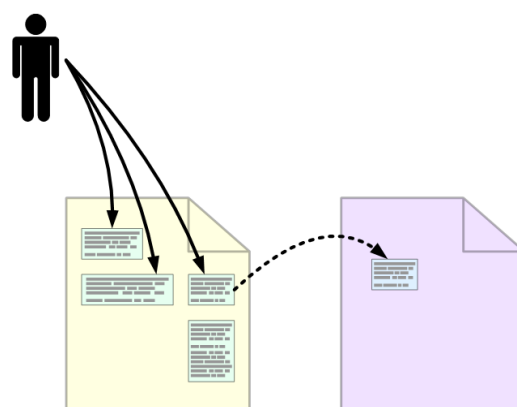
Le classement non strictement hiérarchique est souvent une nécessité au niveau de la publication. L'article traitant d'un match de foot de l'équipe locale pourra être trouvé à la fois dans la rubrique 'sport' et dans la rubrique 'infos locales'. **La possibilité de retrouver un article en différents points de l'arborescence permet de satisfaire des visiteurs qui ont chacun une logique particulière ; on parle alors de multi-positionnement.**



En effet, et c'est un point fondamental, il ne suffit pas de présenter son contenu de manière logique et organisée, il faut aussi **que cette logique soit celle du visiteur**. Or tous les visiteurs ne raisonnent pas à l'identique, et il faut donc aussi permettre de retrouver un contenu selon différentes logiques.

Pages et contenus

Pour certains CMS, un contenu est toujours créé *dans une page spécifique*, voire même à *une position spécifique dans la page*. Pour d'autres, la création d'un contenu, sa validation, son versionning, sont indépendants de la notion de page. Ce n'est qu'une fois créé et validé que le contenu peut être *placé* dans une ou plusieurs pages. Cette approche découle du principe de séparation de la *gestion* (en amont) et de la *publication* (en aval).

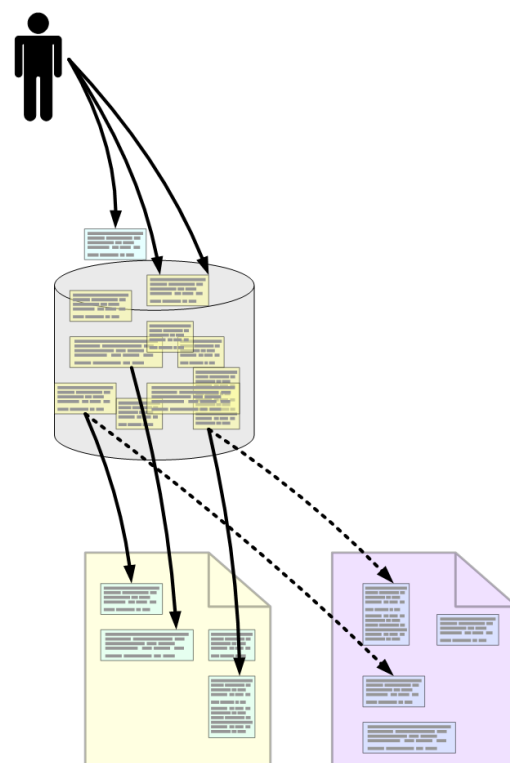


Même lorsqu'un contenu est placé *en premier lieu* dans une page, il est important qu'il puisse être restitué dans d'autres pages sans pour autant être dupliqué. Ainsi une mise à jour du contenu impactera automatiquement les différentes pages dans lesquelles il est publié.

Cette restitution multiple d'un contenu, parfois appelée *multi-positionnement* des articles, peut être opérée par un positionnement explicite : le gestionnaire place l'article dans telle et telle pages, ou bien elle peut s'envisager au travers de *règles* régissant la publication (à partir de la taxonomie pour certains).

Typiquement la présentation des cinq dernières actualités en page d'accueil, ou bien en marge de certaines pages est un exemple de publication par règles. Il n'est pas nécessaire qu'une nouvelle actualité soit *placée* dans telle et telle page (dans la page d'accueil dans notre cas), c'est parce qu'elle est de type *actualité* (ou qu'elle a un mot clé Actualité associé) et parce qu'elle est récente, qu'elle y figurera automatiquement.

Certaines solutions se basent sur le multi-emplacement pour gérer du multi-site ; cela correspond à l'utilisation d'un même contenu au sein de différents sites (dans ce cas, souvent, un seul outil de gestion avec une base de données unique est utilisé).



Spip, Drupal et Joomla sont orientés ‘contenus’, chaque contenu (article, brève, rubrique, ...) formant une page, avec une url. Les mots-clés permettent ensuite de ventiler les contenus sur plusieurs pages.

Typo3 est orienté ‘pages’. Une page contient une ou plusieurs colonnes, et chaque colonne contient un ou plusieurs contenus ou modules, ce qui donne une grande souplesse de mise en page. La validation et le suivi des versions se font au niveau de la page dans son ensemble.

Jahia est lui aussi orienté ‘pages’, reflétant là son aspect portail. La mise en forme des pages est souple, dépendant des gabarits réalisés. La catégorisation bien implémentée – ou encore le ‘piqueur de contenus’ (amélioré sur la version 6) - permet ensuite de ventiler les contenus sur plusieurs pages. La validation et le suivi des versions sont réalisés par page et contenu. Jahia permet également de faire du multi positionnement de contenu à travers le principe de « liens virtuels ».

eZ Publish est davantage orienté ‘contenus’. Les contenus suivent le processus de validation, sont suivis en versions, sont placés dans plusieurs dossiers, qui sont eux-mêmes des contenus. Un contenu a une url à travers son positionnement dans l’arborescence. La gestion des pages se fait au travers des templates. A noter l’existence d’eZ Flow, une extension qui permet de construire ses mises en page à partir d’éléments du référenciel (contenus).

InfoGlue a une orientation ‘contenus’. Les contenus sont validés, suivis en versions et multi-positionnés par l’intermédiaire de composants. Infoglue sépare l’arborescence de contenu de celle du rendu. Les contenus sont visibles sur le front office par l’intermédiaire de composants posés dans les pages par le webmaster.

Versions d’un contenu

La gestion de version et la capacité à conserver les versions antérieures d’un même article (d’un contenu au sens large), font parties des fonctions de base d’un CMS. Elles s’appliquent à la fois aux versions anciennes et aux versions futures. Un même article peut ainsi avoir une version en ligne, une version N+1, validée mais non publiée, et une version N+2 à l’état de brouillon.

Comme nous le verrons un peu plus loin dans ce document, la possibilité d'avoir plusieurs versions simultanées d'un même contenu est nécessaire au fonctionnement des workflows de validation. C'est ce qui nous permettra de suivre des circuits de validation pour chaque modification, et donc de toujours s'assurer de la qualité des publications par un responsable.

La gestion des versions peut se faire au niveau de chaque contenu, mais aussi au niveau des pages – une page présentant un ou plusieurs contenus agencés –, ou encore au niveau d'une partie d'un site, voir au niveau du site complet. La gestion de version de site est complexe à manipuler, mais elle est intéressante pour publier conjointement des ensembles de contenus, répartis sur tout le site. C'est une pratique rencontrée dans les grosses structures où une unique communication peut entraîner un lot de changement ; dans ce cas, les modifications ne peuvent être publiées individuellement ; elles forment un ensemble cohérent qui doit être « manipulé » d'un bloc.

Qui parle de gestion des versions parle de gestion des conflits. Aussi, l'outil doit éviter toute perte, en interdisant l'accès à deux personnes simultanément pour le même document, ou en créant deux versions distinctes, une par personne.

Dans un souci de traçabilité, les contributeurs ont besoin d'une part de savoir qui a modifié quoi sur le site, et d'autre part, de connaître les différences entre deux versions, que ce soit au niveau d'un contenu, d'une page, voire d'un site complet.

Un point différenciant entre les outils est la capacité à gérer les versions des fichiers joints, comme les fichiers PDF par exemple, mais aussi les images. Car quand un contributeur modifie un communiqué de presse avec un fichier PDF joint, en remplaçant le texte mais aussi le PDF, il faut que l'outil différencie les deux versions du PDF, pour le circuit de validation et pour un possible retour en arrière.

Joomla n'intègre pas de système de gestion de versions. Il verrouille cependant les articles en cours d'édition afin d'éviter les conflits. Impossible donc de revenir en arrière sauf à dupliquer une page et à la déclarer en archive.

Drupal intègre désormais un gestionnaire de versions. Cependant la création d'une nouvelle version est déclarative ; c'est le contributeur qui décide de la création d'une nouvelle version. Cela peut être contourné mais le système de gestion de versions sous Drupal reste relativement léger.

Spip conserve les versions des articles, et sait afficher les différences entre deux versions ; mais la gestion reste très légère. Spip verrouille les articles lorsqu'un utilisateur les modifie, ce qui évite tout conflit. Le point faible de Spip est qu'il ne permet pas de modifier un article indépendamment de la version en ligne.

Typo3 permet de gérer nativement des versions d'arborescences complètes grâce à sa notion de workspace. Depuis la version 3.7, un module de versionning permet de créer des nouvelles versions de pages, mais aussi d'arborescences complètes, de les travailler, puis de les basculer en ligne. Le différentiel reste au niveau de chaque contenu. Ainsi, un contributeur qui travaille sur un contenu du workspace Draft (brouillon) et le valide, bascule automatiquement ce contenu dans l'environnement Live (site en ligne). Très pratique.

OpenCms gère les versions des contenus. La séparation des espaces de travail et de publication permet de travailler une arborescence, un contenu ou un fichier sans impacter la version en ligne.

eZ Publish sait gérer les versions de contenus, y compris les versions des fichiers associés aux contenus ; il est très complet sur cet aspect. eZ Publish associe une version à un état dans le circuit de validation, ce qui permet de maîtriser parfaitement les contenus, en modification et en suivi. eZ Publish offre un système de comparaison de version.

Jahia dispose d'un module de suivi des versions. La modification d'une page ou d'un de ses contenus, puis sa validation se font en parallèle de la version en ligne, avec même une prévisualisation de l'ensemble du site, dans sa version de travail ; très intéressant. Jahia propose un système de comparaison de version.

InfoGlue gère les versions au niveau des contenus, et permet d'avoir pour un même contenu un état publié et un état brouillon. Lorsqu'un contenu est publié, son affichage dépend de son cycle de vie (date de vie, date d'expiration), cela introduit un nouvel état que l'on pourrait nommer « archivé ». Le contenu est toujours présent dans le back office mais pas visible dans le front. Infoglue permet la publication par lot.

Manipuler le contenu

Cette rubrique traite de la manipulation des contenus, y compris la gestion des cycles de vie. Après avoir défini un type de contenu, on peut lui donner vie, c'est-à-dire créer, enrichir, modifier des contenus, et les publier. C'est bien sûr la fonction centrale des CMS, et elle nécessite une attention toute particulière.

Interface de gestion

Au fil des années, l'interface de gestion des contenus est devenue un élément clé de l'utilisation des CMS.

Les contributeurs et les administrateurs y passent le plus clair de leur temps en période de mise à jour. Et c'est pourquoi, il est important qu'ils disposent d'une interface efficace qui facilite le travail et permette un gain de temps.

L'utilisation de technologies comme l'Ajax (ensemble de technologies qui permet l'échange et la manipulation de données sans le rechargement de la page) permet de mettre en œuvre des interfaces qui n'ont plus rien à voir avec des simples pages Web ; on peut désormais construire de véritables applications interactives utilisables à travers un simple navigateur. Et on peut aller très loin : utilisation de raccourcis claviers, sélection multiple, « compléteur » de champs, « drag and drop », menu contextuel, etc.

Il est plus facile de supprimer une page d'un simple clic à partir d'un menu contextuel que de charger cette page, de chercher le bouton « Supprimer » en bas de page et de cliquer. Il est plus facile d'associer des mots clés à l'aide d'un « compléteur » (qui à partir de quelques lettres vous proposera les mots clés existants) que de les saisir un par un (sans fautes). Etc.

Au delà de la couche ergonomique (qui reste un point fondamental d'une bonne interface de gestion), la facilité d'utilisation est capitale dans l'appropriation de l'outil. Trop d'outils font encore l'objet de rejet de la part des contributeurs, qui pour la plupart ne sont pas des informaticiens. D'ailleurs, on s'aperçoit que les CMS les plus utilisés sont les plus faciles d'accès et de prise en main.

Notons que certains CMS aux interfaces complexes mettent en avant le côté « productivité » de leur back office, c'est-à-dire qu'à défaut d'être intuitifs, ils sont très efficaces après prise en main.

Naturellement, l'interface de gestion n'est que l'un des critères de choix, qui ne doit pas faire oublier les autres : extensibilité, performances, etc.

A ses débuts, **Joomla** a percé rapidement notamment grâce à la qualité de son interface de gestion. Très graphique et soignée, ses écrans ont inspiré beaucoup d'outils informatiques Web. Son interface est facile à prendre en main même si certaines terminologies particulières peuvent être difficiles à comprendre au premier abord (menu, section, catégorie). En 2010, elle apparaît comme rudimentaire car quasiment dépourvue d'Ajax (pas de drag and drop pour ordonner les pages par exemple).

Drupal est aussi très simple d'utilisation. Ses interfaces sont sobres et vont à l'essentiel. Drupal se différencie par son interface de gestion (back office) dans le ... front office. Cela peut paraître assez déroutant de premier abord mais très intéressant à l'usage. Seul bémol, sa navigation complexe entre les fonctionnalités lorsqu'on active de nombreuses extensions. La version 7.0, récemment disponible, apporte une meilleure ergonomie avec notamment une forte utilisation de l'Ajax. Prometteur.

L'interface de **Spip** a peu évolué, graphiquement notamment, au fil des versions. Ses utilisateurs l'apprécient pour sa facilité de prise en main. La version 2.0 a encore amélioré cet aspect et en fait un des CMS les plus faciles d'utilisation. On regrettera toutefois son éditeur « wysiwyg » par raccourcis typographiques qui oblige parfois les contributeurs à quelques contorsions intellectuelles.

Typo3 a peu de points faibles mais son interface de gestion en fait partie. Au-delà de la « philosophie » de l'outil, complexe mais redoutablement efficace, ses écrans sont difficiles à lire submergés par les nombreuses fonctionnalités. Heureusement, la version 4.3 apporte une contribution frontend-editing (depuis l'interface de front office – pour rappel, partie visible des internautes) bien faite qui facilite la contribution pour des publics sans compétences techniques. La future version, la très attendue version 5, devrait apporter une nouvelle interface graphique beaucoup plus ergonomique. En attendant, les utilisateurs de Typo3 mettent en avant la productivité liée à son utilisation.

OpenCMS a la particularité de « reproduire » des écrans de Windows. Cela peut représenter une force dans la conduite du changement qu'entraînent certains projets. Pourtant le mimétisme pousse à des comportements surprenants dans des interfaces Web (double ascenseur par exemple).

eZ Publish dispose d'une interface relativement sobre, claire à lire et facile d'accès. Elle propose un bon compromis entre fonctionnalités avancées et usabilité. La nouvelle version sortie récemment, la 4.3, apporte une interface refondue, plus neutre et surtout dotée de fonctionnalités Ajax très pratiques.

Jahia propose une interface soignée et travaillée pour laquelle on peut difficilement trouver quelque chose à redire. Son frontend editing est impressionnant de facilité ; une garantie pour la prise en main des utilisateurs. Seul reproche, lié à son architecture par blocs (une page est constituée d'un ensemble de blocs), on trouve parfois difficilement les points d'entrée vers les interfaces de modification ; idem pour les états de workflow.

Infoglué sépare clairement l'étape de contribution de l'étape de mise en page ; ce qui est un point fort. Ses interfaces sont agréables mais dépourvues d'Ajx. Les contributeurs peuvent travailler uniquement sur la production de contenu sans se préoccuper du rendu ou du positionnement final. Le positionnement d'un contenu dans l'arborescence de rendu peut être réalisé automatiquement ou manuellement par le webmaster. La construction d'une page en partant d'une page blanche peut paraître un peu complexe pour un novice mais se révèle finalement très simple après quelques essais.

Dépôt d'un contenu

Dans les premières versions des outils de gestion de contenu, l'intégration d'un contenu se faisait exclusivement par un simple formulaire (exemple : titre, sous titre, texte, etc.) sans même un éditeur Wysiwyg.

Aujourd'hui les choses ont bien changé, et il existe une multitude de façons d'incorporer un contenu : formulaire avec éditeur graphique, multi-upload, webdav, frontend editing, intégration Microsoft Office ou Open Office, email, ...

Formulaires et éditeurs

Les formulaires restent la méthode principale d'intégration de contenus dans un site Web.

Aujourd'hui, ils sont plus complets et plus efficaces.

Premièrement, ils incluent presque tous un éditeur graphique (Wysiwyg : What you see is what you get). Pour bien faire, cet éditeur doit être complètement intégré à votre CMS (voir plus haut).

Deuxièmement, les formulaires de saisie (à l'image des contenus) sont plus structurés (voir l'exemple plus haut de « La recette de cuisine ») ce qui permet d'assister le contributeur dans la saisie de son contenu.

Frontend Editing

Le Frontend Editing consiste à modifier un contenu directement depuis le front office (c'est-à-dire depuis la partie visible des internautes). On travaille toujours à partir de formulaires mais la navigation et la recherche d'une page à modifier est plus aisée. Le Frontend Editing est particulièrement adapté à la relecture ou à la modification d'un site, en parcourant les pages et en s'arrêtant sur celles qui méritent une modification. Il s'agit d'un mode de contribution très simple grâce, notamment, à des interfaces dépourvues de fonctions avancées.

Multi-upload

Des champs Upload (chargement de fichiers) ont toujours existé dans les formulaires ; cependant, les chargements se faisaient de façon unitaire – un par un. Aujourd'hui, de nouveaux composants (en flash souvent) permettent de charger simultanément X fichiers d'un coup. Quelle révolution lors de l'initialisation d'une médiathèque !

Webdav

Le protocole Http s'est enrichi d'une surcouche permettant de faire d'un serveur web un serveur de fichiers. Le Webdav (Web-based Distributed Authoring and Versioning) permet à des utilisateurs d'accéder et de modifier des fichiers (ou des contenus au sens large pour certains) sur des serveurs web distants.

Par rapport au Ftp bien connu, le « protocole » Webdav permet en plus de verrouiller les fichiers lorsqu'ils sont modifiés par un utilisateur, mais surtout, il évite de devoir faire une copie locale du fichier avant de le modifier. **Les utilisateurs peuvent alors travailler en collaboration sur les fichiers du serveur distant, comme s'il s'agissait d'un simple répertoire partagé sur le réseau.**

Les postes des utilisateurs doivent être équipés d'un client Webdav leur permettant d'y accéder. L'explorateur de fichiers de Windows en est équipé, ainsi que Microsoft Word et Open Office, et il existe plusieurs clients disponibles (et gratuits) pour Linux ou Mac.

L'intégration de Webdav aux outils de gestion de contenus ouvre de nouvelles fonctionnalités : accès aux fichiers publiés et processus de publication hors de l'interface web.

Notons que l'accès Webdav à un CMS doit mettre en œuvre la même sécurité que l'accès interface web, avec les mêmes utilisateurs et droits.

Intégration avec les traitements de texte

Pour beaucoup d'entreprises et organisations, la création d'un site web passe par une interminable étape de copier/coller. En effet, la majorité des contenus sont déjà disponibles sous des formats bureautiques et précieusement conservés par le service de communication.

Aujourd'hui les meilleurs CMS proposent des interfaces avec les principaux traitements de texte du marché. Associée au protocole Webdav, l'intégration est terriblement efficace ! Il vous suffit de déposer vos fichiers via Webdav et ils sont automatiquement convertis en article sur votre site Web. A terme, vous pourrez gérer votre site web (le contenu du moins) directement depuis votre traitement de texte favori.

Intégration d'emails

Les éditeurs et les communautés ont bien compris quels étaient les principaux freins à l'utilisation de leurs solutions. Ils ont donc cherché à multiplier la façon d'intégrer du contenu en toute simplicité.

Parmi ces méthodes, on trouve désormais l'intégration d'emails directement dans le CMS. Le principe : le parcours d'un compte de messagerie et la transformation des messages en article ou brèves (avec pour les plus avancés, la prise en compte des pièces jointes).

Cette méthode a le mérite d'être très simple d'usage. On la recommandera en mode Intranet pour collecter, par exemple, des actualités sur des sujets divers et éclatés. L'intégration d'emails doit être couplée à un workflow pour valider les contenus avant la mise en ligne.

Un site qui fonctionne est un site mis à jour ! Cette dimension est importante, c'est pourquoi nous considérons qu'il s'agit d'un concept fondamental de la gestion de contenu. Comme nous l'avons vu, il existe un large panel de méthodes d'intégration de contenu ; soyez vigilant sur ce point selon le format et l'organisation de vos contenus.

OpenCMS permet l'intégration de contenu via formulaires structurés, webdav et frontend editing.

Typo3 propose l'intégration de contenu de différentes manières : frontend editing, formulaires adaptés, multi-upload ainsi que email par extension.

Joomla intègre simplement les contenus par formulaire.

eZ Publish est très complet du point de vue de l'intégration des contenus. En effet, il est possible d'intégrer des contenus à l'aide : de formulaires structurés (souvent accompagné d'éditeur wysiwyg), via OpenOffice, via un composant flash de chargement multiple et via l'implémentation de webdav. eZ Publish propose un mode frontend editing ainsi qu'un module d'enregistrement de contenu par l'intermédiaire du mail.

Spip est plus limité sur ce point. Il propose des interfaces figées pour intégrer les types de contenu proposés (article, brève, rubrique) et le chargement multiple par l'intermédiaire d'un composant de la communauté.

Jahia propose l'intégration des contenus par formulaires structurés, webdav, multi-upload par l'intermédiaire de fichiers Zip et frontend editing.

Infoglue intègre les contenus par formulaires structurés et multi-upload (via un composant flash). Il dispose également d'une sorte de frontend editing via le back office ...

Drupal permet l'intégration de contenu par l'intermédiaire de formulaire adapté. Il supporte également webdav et le chargement multiple par l'intermédiaire de plugins. Le frontend editing est natif ; il s'agit d'ailleurs du parti-pris de Drupal.

Cycle de vie d'un contenu

Au sein de la base des contenus, les articles naissent, vivent et meurent parfois. Voyons comment les CMS gèrent ce cycle de vie.

L'un des besoins les plus classiques consiste à préparer un article *avant* sa date de publication, et de programmer sa mise en ligne automatique à une date donnée.

Un article peut également avoir une durée de vie connue à l'avance: soit en terme de durée, soit selon une date limite. On peut ainsi créer un article intitulé *"comment remplir votre déclaration d'impôts ?"*, et indiquer dès sa création que cet article est valable un an, ou bien jusqu'au 31 décembre 2011.

Que se passe-t-il alors à cette date ? L'article peut simplement être retiré de la publication de manière automatique. Dans certains cas, le contributeur souhaiterait être prévenu à l'avance de la péremption prochaine. Le CMS pourrait donc, dès la création de l'article, permettre de définir sa date de péremption, mais aussi si le contributeur souhaite être prévenu, combien de jours à l'avance, et si le webmaster également doit être prévenu. Une fois averti, le contributeur peut mettre à jour l'article si besoin, et définir une nouvelle date de péremption.

Ce sont là des cycles de vie relativement simples. On peut souhaiter des choses plus sophistiquées, et en particulier qu'entre sa naissance et sa mort, les modalités d'apparition d'un article changent à certaines échéances. Il peut ainsi rester une semaine sur la home page, puis trois mois dans la rubrique actualités, puis un an dans la rubrique archives, et finalement disparaître.

Pour mettre en œuvre ces cycles de vie, on s'appuie essentiellement sur des déclencheurs et des workflows.

Par rapport à cette gestion de cycles de vie, **Joomla** permet uniquement la définition de dates de début et de fin de vie des articles.

Drupal dispose d'un module « Scheduler » permettant de gérer le cycle de vie des contenus.

Spip répond au besoin standard et permet de publier des articles, préparés à l'avance, à partir d'une date donnée, mais sans date de fin de publication. Une légère modification (en développement) permet de mettre en place une date de début et une date de fin de publication.

De la même manière, **Typo3** définit optionnellement une date de début et une date de fin de visibilité pour chaque article, sans alerte de fin de vie. Typo3 gère aussi la durée de vie sur une page, et même sur une arborescence complète.

eZ Publish n'associe pas la durée de vie à un contenu, mais cette fonctionnalité est rendue possible par simple configuration de l'outil : il faut ajouter les 2 dates (ou plus) à chaque structure de contenu, et configurer l'événement de workflow pour qu'il compare la date du jour avec la date de début et la date de fin saisies. Cela n'est pas trivial mais fait partie de la philosophie de l'outil. A noter qu'il existe une extension eZ Flow qui permet de composer les pages par blocs et de les animer (rotation automatique, publication le .., etc.).

Jahia permet de définir une date de début et une date de fin pour n'importe quel contenu. Il est également possible de planifier une publication journalière (par exemple, tous les jours de 17H à 18H) et « hebdomadaire » (par exemple, le lundi de 9H à 10H, le mardi de 15H à 16H, etc.). Pour des opérations plus sophistiquées, il est nécessaire de mettre en œuvre et de paramétrer le moteur de workflow.

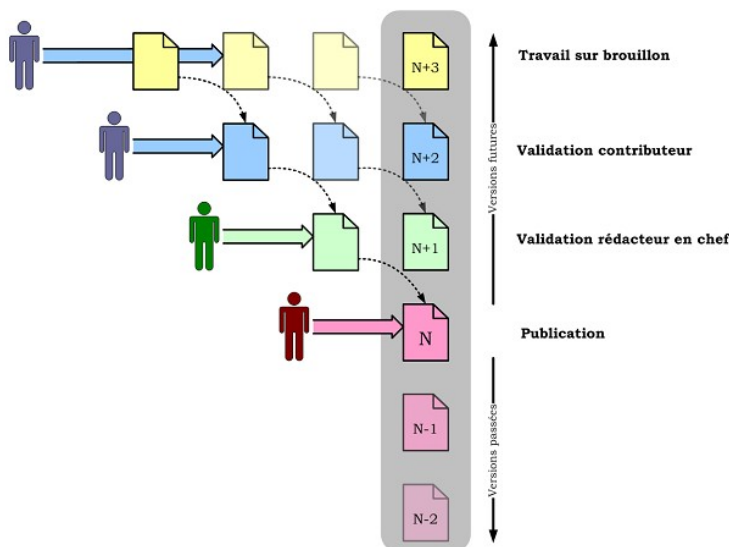
Dans les métadonnées d'un contenu, **InfoGlue** propose une date de début et de fin, ce qui permet de définir un cycle de vie pour chacun des contenus.

Workflow de validation

Le besoin apparaît souvent de pouvoir séparer les étapes de contribution et de validation. C'est le cas en particulier dans des contextes où la contribution est décentralisée, déléguée à un nombre important d'intervenants. Si l'on souhaite malgré tout garder un contrôle sur les contenus, et c'est souvent fondamental, il faut qu'un article soit d'abord rédigé, puis dans un second temps validé et éventuellement mis en ligne. Cela est d'autant plus important que les contributeurs ne sont pas toujours des spécialistes, ni de la communication, ni du web.

Dans certains contextes, ce sont même plusieurs échelons de validation qui sont nécessaires : un contributeur écrit, son chef de service valide à son niveau, et le responsable de la communication valide une seconde fois.

La validation doit donc être mise en œuvre par le CMS avec toutes les



caractéristiques d'un véritable workflow : les intervenants doivent être avertis par mail des tâches qui leur incombent, et doivent d'un simple clic pouvoir consulter la liste des articles qui attendent leur validation.

Mais ce n'est pas tout, une fois un contenu créé, validé puis publié suivant le circuit de validation, il doit pouvoir être modifié, validé puis publié, pendant que ce contenu en ligne reste inchangé. A un circuit de validation, il faut donc pouvoir associer une gestion avancée des versions, permettant d'avoir plusieurs états simultanés pour un même contenu.

Les workflows sont donc des éléments à part entière du cycle de vie d'un contenu. Ils doivent être considérés à leur juste importance en fonction de votre organisation de travail. Et ceci d'autant plus que l'on observe des mises en place plus ou moins complètes selon les CMS testés.

Avec **Spip**, un article peut être à l'état *brouillon*, *proposé*, *publié*, *refusé* et *supprimé*. Selon son statut de rédacteur ou d'administrateur, on pourra ou ne pourra pas mettre en ligne un article. Le workflow mis en place est donc assez limité, mais suffisant dans bien des cas. Notons que pour modifier un article déjà en ligne, il faudra le dupliquer pour passer ensuite par le circuit de validation, un contenu Spip ne pouvant avoir qu'un état à la fois.

Pour **Joomla**, la gestion de la validation est très simple. Un contenu est dans un état « non publié » ou « publié ». Seul un utilisateur disposant de suffisamment de droits pourra passer un document dans l'état publié. Joomla n'est pas adapté à un mode de production collaboratif des contenus.

Drupal n'inclut pas nativement de gestion de workflows. Pourtant, il est à noter qu'il existe un très bon module additionnel pour cela. Les paramétrages initiaux peuvent paraître assez lourds mais les possibilités finales sont très nombreuses.

Typo3 intègre depuis sa version 4 la notion de *workspaces* qui permet de travailler sur plusieurs environnement (une 'préprod', une version 2011, une recette, ...) puis de basculer en production une page ou une arborescence de pages. Pour chaque workspace, on définit trois groupes : contributeurs, relecteurs et valideurs, correspondant à un workflow de publication à 3 niveaux, spécifique au workspace. Au final, le valideur décide de la publication dans le workspace « live ». Il est possible d'aller plus loin en cumulant les workspaces.

OpenCms, de son côté, offre un dispositif de workflow qui est à la fois extrêmement générique, et pratiquement décorrélé du processus de validation proprement dit : il permet de construire toute sorte d'enchaînement de tâches, mais il n'y a pas de lien précis entre ces tâches et la gestion de contenu. L'avantage est que l'on peut construire ainsi un workflow dans lequel on demande à tel intervenant de compléter tel article, ou de ranger son bureau. L'inconvénient est que la valeur ajoutée de l'outil est faible par rapport à un simple échange de mail.

eZ Publish permet de définir des workflows associés à des sections, appliqués ou non selon les utilisateurs et groupes. Cette fonctionnalité permet de mettre en œuvre des workflows à un ou plusieurs niveaux de validation. Un système de notification envoie automatiquement des mails d'alertes et de demandes de validation aux intéressés. De plus, eZ Publish permet d'avoir simultanément autant de versions que l'on souhaite d'un même contenu : 3 brouillons, 1 en cours de validation, 1 publié, 5 archivés ... ce qui évite tout conflit dans les mises à jour par les différents intervenants. L'ajout récente des « States » (états) a augmenté les possibilités des workflow d'eZ Publish.

Jahia, depuis sa version 5.0, permet de mettre en œuvre des workflows multi-étages et de les affecter aux contenus selon une logique d'héritage. Un nouveau workflow peut être défini de manière interactive, au sein du back-office. Il existe par défaut quelques workflows types (0, 2, 3 et 4 niveaux) qui répondent à la majorité des cas et qui se mettent en œuvre très simplement. Une utilisation très facile et très agréable.

InfoGlue intègre de base des workflow de publication (contribution, validation, etc.). Il embarque également l'outil de workflow open source OpenSymphony, qui permet de définir en Xml des workflows complexes. Ces workflows paramétrables peuvent être appliqués aux contenus, mais aussi être couplés à d'autres applications de l'entreprise, par le biais de web services. La gestion des versions est ici bien traitée, avec d'une part la possibilité de modifier un contenu et de suivre le workflow en laissant le contenu en ligne inchangé, et d'autre part la possibilité de publier des contenus par paquets, ce qui permet d'obtenir des versions de sites cohérentes, publiables et dépubliables.

Contribution collaborative

Les outils de gestion de contenu ont pour principaux objectifs, la production et la publication de contenu.

La création d'un contenu se fait au travers d'un cycle de vie : étape par étape. Elle se fait aussi, le plus souvent, grâce au travail collaboratif, au-delà de la simple validation.

Ne dit-on pas que les meilleurs résultats sont collaboratifs ?

Selon que vous ayez déjà des contenus formalisés (terminés) ou que vous ayez à les produire, vous devrez apporter ou non une attention particulière à la notion de contribution collaborative.

La collaboration peut prendre différentes formes : commentaires sur les articles, forum attaché, calendrier partagé avec gestion des tâches, wiki, blog, révision de document avec suivi des modifications, notations, annotations, etc.

La notion de collaboration doit être rapprochée de la notion de workflow ; mais il peut exister des espaces de travail isolés qui permettent de concevoir un contenu.

Les CMS qui incluent cette dimension sont rares ; c'est pourtant un axe attendu. Aujourd'hui, on sépare de plus en plus les communications selon les médias (on ne communique pas de la même façon sur un format papier ou sur un format Web). Les CMS deviennent donc un lieu de production de contenu et doivent bénéficier d'outils de collaboration.

Pour la collaboration des contributeurs, **Joomla** propose uniquement un système de messagerie équivalent à un « mini-webmail ». Cela permet d'échanger des messages entre les utilisateurs.

Drupal ne permet pas nativement de travailler collectivement sur de la production de contenu. Le travail collaboratif n'existe pas ; mais c'est sans compter sur les nombreuses extensions du produit qui peuvent rapidement transformer votre plateforme en véritable communauté. Reste à trouver les bons modules, les configurer et à assembler le tout correctement. Pas si simple.

Spip fait sûrement partie des premiers CMS ayant introduit le travail collaboratif. Même s'il reste une marge de progression intéressante, il offre déjà un beau périmètre : proposition d'article à l'évaluation des autres contributeurs, commentaires et forums entre contributeurs pour un article donné, calendrier, pense-bête, messagerie, annonces, etc.

Typo3 propose nativement peu de fonctionnalités collaboratives coté back office, seul le principe de Workspace est à souligner. Des extensions permettent de mettre en œuvre des tableaux de bord, des messageries et des ToDoList (tâches).

Sur le volet collaboratif, **OpenCMS** se fait remarquer avec son "task management" (centre des tâches). Il s'agit d'une interface qui permet de déléguer des missions sur les contenus (pour qui, pour quand, priorité, commentaire, etc.).

eZ Publish est relativement léger sur l'aspect collaboratif. Au delà de sa gestion des versions associées aux workflows configurables, peu de fonctionnalités sont disponibles. La sortie de eZ Teamroom apportera sûrement des idées pour le travail collaboratif coté back office. A suivre.

Le constat est le même pour **Jahia** ; de nombreuses fonctionnalités de gestion de contenu mais peu d'outils favorisant le travail collaboratif au delà des workflows. Notons que le front office et le back office étant liés, il est possible de créer des espaces privés, restreints d'accès pour les seuls administrateurs.

Infoglue n'offre pas de fonctions collaboratives au-delà des workflows et des espaces distincts : contenu et publication.

Exploiter le contenu

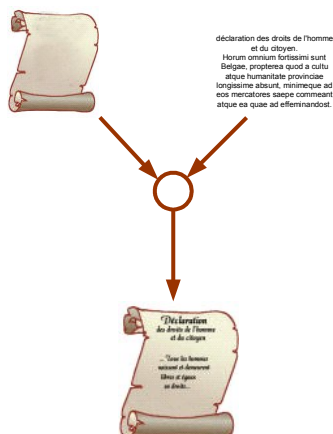
La restitution du contenu (*content delivery*) consiste à produire des pages sur un média donné, en intégrant le contenu au sein d'une mise en forme spécifique.

Le processus de publication peut être décomposé en deux étapes : sélection des contenus et mise en forme des pages sur la base de gabarits.

Gabarits

Mise en forme
(templates)

Contenus



Page publiée

La mise en forme des pages d'un site, pour une restitution Html, est toujours basée sur un dispositif de gabarits, ou *templates*, qui définissent la manière dont les contenus doivent être insérés dans des pages, et quelles indications de mise en forme accompagneront ces contenus.

Pour une page donnée, un ou plusieurs gabarits seront utilisés pour définir à la fois *quels contenus* sont

sélectionnés, où ils sont positionnés dans la page, et *quels attributs de mise en forme* y sont associés. On pourra par exemple définir que telle page ne présentera que les champs *titre* et *accroche* d'une liste d'articles, que ces items seront insérés au sein d'un tableau placé sur la gauche de la page, et enfin que le titre sera en gras tandis que l'accroche sera en italique.

On voit bien sur ce petit exemple, la force de la séparation du contenu et de la mise en forme : il sera toujours temps de changer d'avis et de réagencer la page pour placer ces items à droite, et en couleur bleue, ou de définir une autre page, où chaque item est accompagné également d'une image.

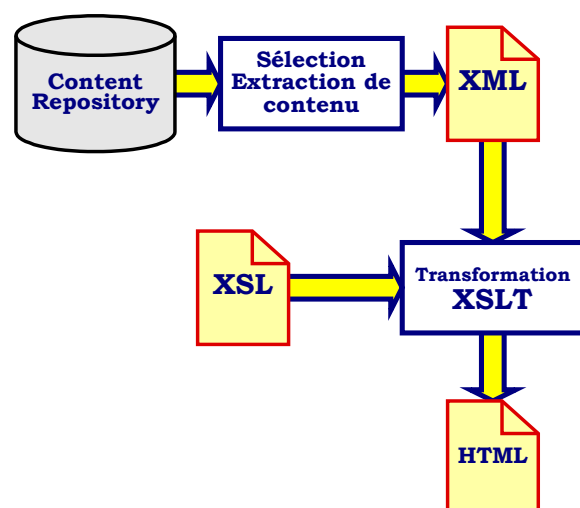
Il existe plusieurs techniques pour mettre en œuvre des gabarits.

La plus simple consiste à insérer dans un fond de page Html des marqueurs particuliers, des balises (en anglais *tags*), qui ne sont pas du Html, mais sont adressées au CMS. Le dispositif d'élaboration de pages du CMS va parcourir le gabarit, analyser ces balises, en déduire les contenus à sélectionner et à insérer à la position de la balise, réaliser l'insertion, puis envoyer la page. Par rapport à ce type de gabarits, il n'existe pas de standard quant à la syntaxe de ces balises, et chaque CMS définira les siennes.

L'avantage de ces premiers gabarits, que l'on appellera gabarits Html, est qu'ils sont pour l'essentiel constitués de Html, et donc assez faciles à manipuler par un webmaster ou un graphiste. **Ils sont cependant limités aux « tags » proposés par le CMS.**

Les pages JSP et PHP sont également une forme de gabarits. Dans le cas des pages JSP, les insertions de contenu peuvent être définies soit par du code Java, soit par des tags faisant référence à des objets Java. Dans ce dernier cas, on parle de librairies de tags, ou *taglibs*. Le grand avantage des gabarits JSP est leur très large adoption pour le développement web Java en général, indépendamment de la problématique CMS. Dans le cas des pages PHP, le contenu est la plupart du temps inséré par du code PHP (il existe des tags particuliers interprétés par le CMS avant l'interpréteur PHP). Comparée à la première méthode, **ces gabarits permettent d'introduire des algorithmes de mise en page complexes** via le langage de programmation (PHP ou JAVA dans nos exemples).

La troisième voie est celle des gabarits Xsl, à base Xml. Dans cette technique, le contenu est d'abord extrait et mis sous la forme d'un objet Xml, qui décrit chaque item de contenu à présenter dans la page. La mise en forme est définie au moyen d'un autre objet Xml, qui fait référence aux items de contenu du premier, et définit la manière dont ces items seront mêlés à des bribes de Html pour constituer la page. On utilise une *transformation Xsl*, qui est un outil standard, pour réaliser l'insertion du contenu, et produire les pages.



La technique des feuilles de style Xsl et des transformations Xslt présente des avantages importants :

- Elle s'appuie sur des standards reconnus, qui s'imposent au delà de la seule problématique de gestion de contenus.
- Elle offre un mécanisme puissant et très modulaire pour constituer les pages.
- Elle ne se limite pas au média Web, et permet aussi d'élaborer des pages destinées à d'autres canaux, en particulier l'élaboration de pages PDF avec l'outil Xml-FOP, qui tend à devenir un standard.

En revanche, cette technique est plus complexe que celle des gabarits Html : le passage de la maquette Html au gabarit Xsl correspondant demande une certaine expertise, et ne pourra pas être confié à un graphiste ou un webmaster. On peut ajouter aussi que l'opération de transformation est relativement gourmande en CPU, et donc lente, mais la plupart des dispositifs de restitution intègrent un mécanisme de cache, de sorte que la transformation ne sera pas exécutée chaque fois qu'une page est demandée.

Ajoutons aussi que, indépendamment du dispositif de gabarits, il existe une technique qui relève strictement du Html pour mettre en œuvre une bonne séparation du contenant et du contenu : les styles CSS. Le principe est de n'indiquer dans le Html que la *nature d'un élément*, et de déporter dans des feuilles de styles toutes les informations de présentation. La page Html dit que « *J. Hunter* » est le *nom de l'auteur*, et la feuille de style indique que le *nom de l'auteur* doit s'écrire avec telle mise en forme. Cette pratique, qui est clairement à généraliser, est indépendante d'un outil CMS en particulier.

Joomla et **Drupal** utilisent des pages PHP qui permettent de créer des gabarits très complets, en utilisant toutes les fonctionnalités de ces CMS. Pour Drupal, on parle de thèmes.

Spip utilise des gabarits Html, avec ses balises spécifiques, mais qui peuvent être complétées le cas échéant par de la programmation PHP.

Typo3 possède également un système de gabarit Html, mais complété de manière intéressante par une configuration interactive (TypoScript), autorisant une certaine latitude de changements sur une simple interface d'administration : changements de couleurs, de fonte, et même d'agencement, peuvent être opérés sans toucher à un gabarit, au moyen d'un simple formulaire Html proposé au back-office.

OpenCms définit un système de gabarits à base de Xml, mais qui n'utilise pas des feuilles de style Xsl. La technique est puissante et bien structurée, mais elle présente l'inconvénient de n'être pas standard, et donc de ne pouvoir bénéficier des outils génériques qui apparaissent autour de ces standards.

eZ Publish utilise des gabarits au format tpl, qui contiennent du code Html et des balises spécifiques à eZ publish. Les tags ont une syntaxe proche du Smarty, et du code Php peut être inclus indirectement pour ajouter des appels externes.

Les gabarits de **Jahia** sont en JSP, ce qui apporte souplesse et performance. Il est à remarquer que ce sont les mêmes gabarits qui permettent la restitution sur le site et l'administration des contenus. Le même gabarit doit disposer des trois différents modes (visualisation, modification et prévisualisation) et de la fonctionnalité de workflow. Il est en général possible de simplement décliner un gabarit, parmi les gabarits disponibles dans l'outil (notion de surcharge).

InfoGlue propose de réaliser les gabarits soit en velocity, un langage de template du projet Apache, soit directement en JSP, ce qui donne plus de souplesse et de possible interaction avec d'autres programmes. La page est une arborescence de composant. Le rendu part des branches les plus profondes. Chaque composant est analysé et traité puis injecté dans le parent. Si un composant contient « <% » au début, il est interprété en tant que JSP ; Velocity sinon (possibilité de faire du freemarker à la place). C'est pourquoi il est possible de mélanger les langages entre chaque composant. On peut imaginer un composant JSP qui génère du code Velocity qui sera interprété par l'itération suivante.

Accessibilité

L'accessibilité se définit par l'accès aux contenus ; par tous quelle que soit la technologie dont dispose l'utilisateur ou son degré d'aptitude (personnes mobiles, seniors, personnes en situation de handicap, etc.). Pour garantir l'accessibilité des sites, des normes techniques ont été établies par la Web Accessibility Initiative (WAI) du World Wide Web Consortium (W3C).

Réaliser un site accessible c'est bien sûr tenir compte des différents handicaps (vue, motricité, etc.), mais également faire en sorte que des critères techniques comme la vitesse de connexion, le type de navigateur ou l'absence de plug-in ne perturbe pas la vision du site ou tout au moins n'empêche pas sa consultation. Il en va de même des connaissances techniques de l'internaute qui ne doivent pas être un frein à la consultation du site, même pour les plus novices.

En plus des bénéfices liés à l'accessibilité pour tous, le respect de ces normes apporte des avantages de taille :

- L'accès au site en mode texte (ex : menu image doublé d'un menu texte caché) et la mise en valeur sémantique des informations (ex : utilisation des balises H1 pour décrire les titres) le rend « compréhensible » aux moteurs de recherche et améliore considérablement le référencement,
- L'utilisation des feuilles de style en lieu et place d'images rend les pages plus légères et donc plus rapides à télécharger, tout en limitant la charge du serveur,
- La différenciation du style et du contenu permet une restitution adaptée pour différents supports (écran / imprimante / voix / etc.) par un simple changement de feuille de style, sans multiplication des gabarits,
- Les modifications ou corrections de mises en forme ne nécessitent pas ou peu d'intervention sur les gabarits, ce qui facilite énormément les opérations de maintenance, mais aussi de recette.

Le respect des normes XHTML/CSS conduit à séparer complètement le fond (contenus XHTML) de la forme (mise en page, couleurs... définies par la feuille de style CSS). Il devient ainsi extrêmement aisé, sur la base du même code de page HTML, d'obtenir des sites graphiquement différents par la simple modification des feuilles de style.

Cela inclut, entre autres :

- l'utilisation de texte en alternative aux images
- l'utilisation appropriée du balisage et des feuilles de style CSS (évitant tous les tableaux imbriqués)
- identifier les entêtes de colonnes de tableaux
- ne pas utiliser les couleurs seules pour passer une information.
- utiliser le minimum de tableaux pour la mise en page

Cependant, l'accessibilité n'est pas seulement une affaire de technique, elle nécessite un traitement tout au long du cycle de vie du site ; par tous les acteurs (développeur, contributeur, administrateur). Les intervenants doivent être sensibilisés pour intégrer correctement leur contenu.

En France, un texte de loi (article 47 de la loi du 11 février 2005) oblige les sites publics à être accessibles ; ceci pour « l'égalité des droits et des chances, la participation et la citoyenneté des personnes handicapées ».

Nous considérons donc que l'accessibilité est un concept fondamental des CMS ; au même titre que la performance ou les fonctionnalités. Même si l'accessibilité n'est pas qu'une notion de technique, les CMS séparant le mieux la forme et le fond obtiennent les meilleurs résultats.

Il est important de différencier l'accessibilité du front office de celle du back office. Les éditeurs et les communautés ont beau faire des efforts, beaucoup d'espace de gestion restent inaccessibles pour certains publics (à cause notamment du javascript).

L'accessibilité du front office dépend essentiellement de trois critères : le montage des gabarits, le code généré par l'éditeur wysiwyg et les possibilités d'intégration de contenu par le contributeur à travers le wysiwyg.

L'accessibilité des gabarits dépend du travail d'intégration – il n'est donc pas différentiant (par rapport aux CMS) à quelques exceptions près (génération de blocs par exemple).

La génération du code par l'éditeur wysiwyg est un point d'attention tout particulier. La séparation du fond et de la forme doit être respectée ; seuls les éditeurs d'**eZ Publish** (XML) et de **Spip** (raccourcis typographiques) y parviennent.

Enfin, les possibilités de l'éditeur wysiwyg sont importantes tout comme l'accompagnement de l'utilisateur lors de la phase de contribution. Pour les tableaux par exemple, l'éditeur doit fournir la possibilité au contributeur d'abrégier les entêtes de colonnes (balise abbr) ; ce n'est pas le cas sur quelques éditeurs testés.

A noter que **Jahia 6** propose nativement un contrôleur d'accessibilité (WCAG). A chaque contribution, le système vérifie la saisie et présente les anomalies ou avertissements WAI. L'administrateur peut passer outre ces alertes.

Sur ce concept fondamental, il n'y a pas de grand gagnant notamment sur la partie front office. Ceux qui s'en sortent le mieux se base sur un éditeur complet au fait des problématiques d'accessibilité et générant un code propre.

Référencement

Le référencement est en partie lié à l'accessibilité ; sur le plan technique du moins. Les optimisations techniques pour l'un favorise l'autre et vice et versa.

Il y a différentes formes de référencement. Nous privilégions le référencement naturel ; celui basé sur vos contenus. Il s'agit de positionner votre site conformément à ses pages, à ses mots. Pour cela, il est nécessaire d'optimiser la mise en place technique des sites.

Pour qu'un site puisse être bien référencé par les moteur de recherche, il est essentiel de garantir une bonne indexation du site ; c'est-à-dire d'ôter tous les freins potentiels au passage des robots.

Chaque page du site peut définir la balise *meta/keywords*, qui permet d'annoncer explicitement les mots clés caractérisant la page. Avec les dispositifs de gabarits, il est toujours possible de définir le contenu de cette balise à partir de l'un des champs décrivant l'article ; mais l'idéal est d'utiliser des attributs dédiés si le CMS le permet. C'est d'autant plus important, que cela permet de fournir des méta-données propres à chaque page.

Autre point, toutes les pages doivent être accessibles sans jamais passer par un formulaire (les moteurs d'indexation ne franchissent jamais un formulaire, même simple) ; et les URL doivent être stables et en particulier ne pas inclure de jeton de session comme le font certains outils de développement.

Les CMS affichent leurs contenus à partir de masques, de gabarits. A ce titre, certains CMS produisent des urls parsemées de paramètres permettant de définir la page appelée. On aura par exemple une url du type www.monsite.fr/article.php3?id_article=18.

Aujourd'hui, il est primordial de disposer d'urls compréhensibles, c'est-à-dire qui donnent une indication quant à la page affichée. Les urls obtenus sont alors du type www.monsite.fr/finances/resultats-financiers.htm pour le contenu 'résultat' de la rubrique 'finances'.

Le deuxième type d'url, porteur de sens, présente l'avantage d'être compréhensible par l'internaute qui sait qu'il est dans la rubrique 'finances' juste en voyant l'url. Ce type d'url présente surtout l'avantage d'être interprété par les moteurs d'indexation tels que Google, qui va donner plus ou moins de points à cette page en fonction des mots présents dans l'url. On parlera d'urls 'propres' ou 'significatives' ou encore d'urls 'Search Engine Friendly'.

Le premier type d'url a l'avantage d'être stable. Si l'article 18 est déplacé dans la rubrique 'finances' à 'résultats', son url restera la même. Alors qu'avec le deuxième type, l'url de l'article sera changée, ce qui peut provoquer des liens cassés dans certains cas.

Tous les CMS s'orientent aujourd'hui vers le deuxième type d'url, même si certains vont plus loin que d'autres.

Deux points nous semblent importants dans la mise en œuvre des urls propres. Tout d'abord, la possibilité pour le contributeur de personnaliser l'url d'un contenu de manière à maîtriser encore plus l'indexation par les moteurs tels que Google. Et ensuite, la gestion par le CMS de la stabilité de l'url. Le CMS doit éviter les liens cassés, en stockant l'historique des urls d'un contenu, et en les maintenant opérationnelles même après que le contenu ait été modifié.

Joomla, comme à son habitude, propose des fonctionnalités "clés en main" pour le référencement. Chaque contenu dispose de métadonnées (description, keywords, robots, author) qui sont automatiquement affichées dans les gabarits. Côté Url, Joomla permet d'activer des Url SEO Friendly mais sans possibilité de sélectionner le format de ces dernières.

Spip a été un des premiers CMS à proposer une alternative aux Url complexes. Aujourd'hui, vous pouvez choisir le type d'Url (friendly) que vous souhaitez depuis l'espace de gestion. Au niveau des métadonnées, Spip propose de les générer automatiquement à partir du contenu du site. Nativement, il n'est donc pas possible d'affiner par page les quelques champs de description Méta. Pour optimiser le référencement, Spip propose des choses intéressantes comme la génération automatique du fichier sitemap.xml (pour Google) ou le fichier robots.txt.

Infoglue propose une gestion des métadonnées par page et des Urls SEO Friendly. L'affichage ou non de ces métadonnées côté front office dépend des composants utilisés. Idem pour le sitemap Google qui est une sorte de plan de site habillé ; un simple composant.

Jahia propose une interface de gestion des métadonnées sur chaque contenu. Il propose également la possibilité de définir manuellement des « clés d'Url » (sorte d'urls propres). Smile a déjà mis en œuvre une génération automatique des urls propres basée sur le nom des pages.

OpenCMS propose une gestion des métadonnées par page et des Urls friendly.

Typo3, comme tous CMS complets, propose une gestion des métadonnées par page et des Urls friendly. Il propose aussi, par l'intermédiaire de plugins, un sitemap Google. Les métadonnées peuvent être générées automatiquement à partir du contenu des pages.

Drupal propose une gestion des Urls propres. Concernant les métadonnées, il fonctionne comme eZ Publish par l'ajout d'attributs complémentaires dans les objets de contenu. Un module est également disponible pour gérer les metas.

eZ Publish permet un bon référencement naturel. Il propose une bonne gestion des métadonnées par l'intermédiaire d'attributs mais cela nécessite un paramétrage ainsi qu'un léger développement au niveau des gabarits (philosophie de l'outil). La gestion des Url est excellente avec le suivi des liens cassés, une gestion multilingue des Url, un principe d'alias, etc.

Mesure d'audience

Tout administrateur de site se doit d'être intéressé par l'audience, tant au global que détaillée pour chaque page. Il est possible que certaines pages ne soient pratiquement jamais vues, soit qu'elles n'intéressent personne, soit qu'elles soient mal mises en valeur, mal référencées, etc.

Suivre l'audience est donc absolument fondamental.

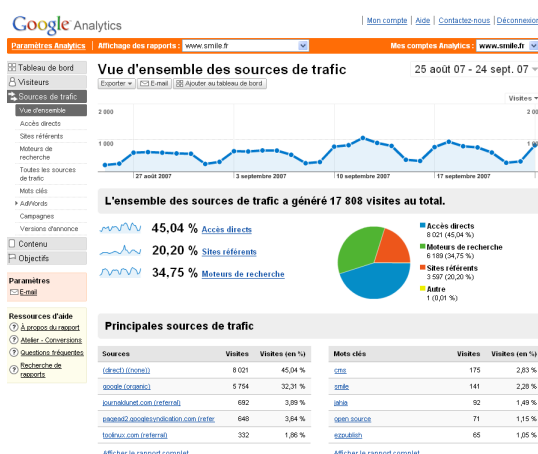
Il existe trois techniques pour le suivi de l'audience :

- L'analyse des logs du serveur web
- Un suivi externalisé, reposant sur des balises intégrées aux gabarits du site
- Une mesure intégrée au CMS lui-même.

L'analyse des logs du serveur est la technique la plus ancienne. Le serveur Http (Apache, IIS, etc.) écrit dans un fichier de log toutes les Url qu'il voit passer. Il existe de nombreux outils qui savent analyser ce fichier et en déduire une grande quantité d'information : nombre de pages vues, nombre de visites, jour par jour, heure par heure, d'où viennent les visiteurs, quel navigateur ils utilisent, quels pages sont les plus visités, et bien d'autres choses encore.

On peut citer de nombreux outils open source de qualité, tel que AWStats. A noter qu'en hébergement mutualisé, cette analyse est généralement fournie par l'hébergeur.

Le suivi externalisé est une technique plus récente, proposée par des prestataires spécialisés, tels que eStat ou Xiti, ainsi que Google Analytics, qui est gratuit et de bonne qualité. Elle consiste à placer dans les pages des balises spécifiques, invisibles au visiteur, et qui provoqueront des hits vers un serveur externe, spécialisé dans la mesure d'audience.



L'administrateur peut se connecter au site du service pour consulter de nombreux tableaux d'analyse de son audience. La forte concurrence entre les solutions amène une grande richesse dans les états disponibles. Cette technique est donc puissante, mais d'une part elle présente un coût récurrent à payer au prestataire (sauf

pour Google Analytics à ce jour), et d'autre part elle implique le marquage des pages, qui peut être fait par l'intermédiaire des gabarits, mais requiert néanmoins un travail spécifique (très faible s'il est pris en compte dès la conception du site).

A noter également l'existence d'outil comme Piwik ou PHPMyVisites qui permettent de mettre en place un service semblable à Xiti mais sur ses propres serveurs. Il s'agit là d'une alternative particulièrement intéressante ; surtout en mode Intranet où les postes des utilisateurs ne peuvent accéder à l'extérieur (Web).

Enfin quelques CMS relèvent eux-mêmes l'audience au moment où ils sont sollicités pour servir des pages. D'une manière générale, cette technique est beaucoup moins puissante que les deux précédentes : elle fournira beaucoup moins d'information sur les visiteurs et beaucoup moins d'analyses. Elle présente un seul avantage, c'est qu'elle permet simplement d'utiliser la connaissance de l'audience dans la restitution elle-même, par exemple en reprenant sur la page d'accueil les articles les plus consultés du moment. C'est cependant rarement mis en œuvre.

Spip, Joomla, Drupal et Typo3 relèvent eux-mêmes les hits sur les pages servies. Les statistiques (répartition des hits, mais également des informations sur les navigateurs ou les systèmes d'exploitation) sont disponibles directement dans le back office. Typo3 permet d'inclure l'outil Piwik dans son interface.

OpenCms, eZ publish, InfoGlue et Jahia requièrent des dispositifs externes au CMS. Mais comme on l'a indiqué, ces solutions externes offrent des analyses sans comparaison, et sont donc recommandées dans tous les cas.

Syndication

Il est courant qu'un site ait à échanger des contenus avec d'autres sites, soit qu'il reprenne des informations de sites partenaires, soit au contraire qu'il mette à disposition une partie de son contenu.

Le moyen normalisé de ces échanges consiste à mettre à disposition les contenus non plus au sein de pages Html, mais sous une forme Xml, qui permettra au site destinataire d'identifier sans ambiguïté chaque champ d'information.

Du point de vue légal, la récupération, et surtout la republication, du contenu d'autres sites ne peuvent se faire qu'avec l'autorisation du site émetteur, qui est à négocier au cas par cas, ou bien par l'intermédiaire d'entreprises de syndication.

Pour que différents sites puissent se comprendre, il faut qu'ils décrivent leurs contenus selon le même langage. Il existe pour cela un standard : Rss ou Rich Site Summary / Really Simple Syndication, un format Xml, dont l'encadré ci-dessus fournit un exemple.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE rss (View Source for full doctype...)>
- <rss version="0.91">
- <channel>
  <title>Yahoo! News Full Coverage - Technology - Chine
  Search Engine Google</title>
  <link>http://news.yahoo.com/fc?
  tmpl=fc&cid=34&in=tech&cat=internet_portals_and
  <description>Brought to you by myRSS - build your own
  <copyright>Copyright (c) 2002 Innovateer Ltd. See http
  conditions.</copyright>
- <image>
  <title>myRSS</title>
  <url>http://myrss.com/images/logoButton.gif</url>
  <link>http://myrss.com</link>
  <width>88</width>
  <height>31</height>
</image>
- <item>
  <title>Why Did Google Want Blogger?</title>
  <link>http://myrss.com/p/r/w/b/fs53.html</link>
</item>
- <item>
  <title>Wired News</title>
  <link>http://myrss.com/p/e/h/x/3z52.html</link>
</item>
```

Du point de vue technique, la mise à disposition d'informations au format RSS n'est pas très difficile. D'ailleurs la plupart des CMS propose un module tout fait pour gérer ce type de flux. Une autre méthode consiste à utiliser un gabarit spécifique (cela nécessite un développement).

En revanche, l'insertion de contenus externes requiert un mécanisme spécifique pour analyser le flux RSS et l'insérer dans le CMS. Les CMS structurant font ça très bien et proposent même une interface pour « mapper » les données RSS en articles ou contenus de tout genre.

On notera que l'insertion de contenus externes au moyen d'une interface RSS est une des fonctionnalités historiques des portails à base de portlet, tels que Jetspeed ou uPortal, ou encore NetVibes.

Spip est particulièrement efficace en matière de syndication, puisqu'il sait nativement non seulement mettre à disposition un flux RSS, mais aussi intégrer un ou plusieurs flux Rss externes : il suffit de lui fournir l'url du Rss. De plus, Spip peut proposer à un administrateur la validation lien par lien des flux importés.

Joomla et **Drupal** permettent non seulement d'activer facilement des flux RSS pour le site et de configurer leur comportement (authentification, nombre d'éléments, ...), mais également d'agréger des flux externes et de les afficher directement dans les pages du site.

Dans l'environnement **Typo3**, l'import Rss n'est pas natif mais une extension permet d'exporter et d'importer des flux de syndication.

eZ Publish permet de publier des informations dans plusieurs formats, dont le format RSS, ce qui lui permet d'être syndiqué par d'autres sites. L'import de flux RSS est possible, et passera, si on le souhaite, par un processus de workflow avant d'être publié. Notons que eZ publish importe les flux dans des objets de contenus du type que l'on souhaite (le mapping des données/attributs est configurable), ce qui nous permet de leur appliquer toutes les méthodes disponibles (multipositionnement, traduction, workflow, ...), et surtout d'agréger des flux de différents émetteurs.

Jahia dispose d'un module d'import RSS. Les flux sont importés séparément dans la base de contenus. L'export RSS est natif dans la version 6 mais se base toujours sur le principe de gabarit, dans lesquels il faut préciser les contenus à exporter (plus de souplesse).

InfoGlue ne propose pas nativement d'import ou d'export RSS. Par contre, l'import et l'export Xml des contenus est natif, il est donc possible d'adapter le traitement pour le réduire au format RSS.

Personnalisation et animation de rendu

La majorité des sites web présentent le même contenu à tous leurs visiteurs. Il existe toutefois différents cas de figure où l'on souhaite différencier la restitution selon les visiteurs et présenter à chacun ce qu'il souhaite voir, ou ce qu'il est autorisé à voir, à partir d'une base de contenu unique, administrée de manière globale.

On peut distinguer une *personnalisation à la demande du visiteur*, et une *personnalisation définie par les habilitations*.

La première est typiquement la déclinaison du principe de portail à la manière de myYahoo, par lequel chaque internaute spécifie les contenus qui l'intéressent, et peut même configurer l'agencement de la page qui lui présentera ces contenus. Ayant défini ses préférences, chaque internaute sera reconnu, soit par une identification explicite, soit de manière transparente au moyen d'un cookie. Ce type de personnalisation est en particulier mis en œuvre au sein de portails d'entreprise, qui sont évoqués au chapitre suivant.

La personnalisation par habilitations n'est généralement pas gérée au niveau individuel, mais en créant des groupes de visiteurs, définis par leur *profil*. Cette personnalisation suppose bien sûr une véritable identification des internautes, qui permettra de retrouver le profil de chacun d'entre eux.

C'est essentiel en particulier sur des sites Intranets, où l'accès à certaines informations devra être restreint.

Globalement, la personnalisation de rendu est peu mise en œuvre car elle pose des problèmes de performances. Il est plus facile pour un serveur de servir 1 million de fois la même page que 1 million de pages différentes ! Pourtant c'est une attente de plus importante de la part des internautes. Lorsque les contenus sont publics, on privilégiera des constructions de page côté client (avec du javascript par exemple). Lorsque les contenus sont restreints à un utilisateur, on privilégiera, si possible, la construction sur la base de tags SSI ou ESI.

Pour certains portails (comme les médias), on travaille prioritairement sur l'animation des contenus. Certains CMS proposent des modules pour animer des pages en tenant compte de la rotation d'article par exemple. Le CMS se charge alors d'animer le site en variant les contenus (sur la page d'accueil par exemple) selon les règles de gestion définies préalablement.

Joomla propose une personnalisation par habilitations sur 3 niveaux d'accès : public (tout le monde), membre (utilisateurs disposant d'un compte), et spécial (utilisateurs disposant d'un droit d'édition).

Drupal ne gère nativement que 2 niveaux d'accès pour la restitution : public et membre. Des modules supplémentaires apportent une plus grande souplesse et permettent de définir des droits détaillés pour chaque contenu. Pour la mise en œuvre de page personnalisée type iGoogle, Drupal dispose du module Homebox.

Spip ne prévoit pas de restitution personnalisée. Il arrive que cette limitation soit rédhibitoire.

Typo3 gère de manière très complète les habilitations en lecture sur le site et permet donc une restitution personnalisée. En fait, c'est le même dispositif qui s'applique aux contributions et aux visites sur le site. Chaque page du site peut être affectée à un groupe d'utilisateurs et dans ce cas ne sera visible que par les visiteurs appartenant à ce groupe.

eZ Publish gère les droits d'accès en restitution et la gestion des comptes utilisateurs. Le forum et les abonnements sont des fonctions qui relèvent de la personnalisation et qui sont mises en œuvre nativement. Notons également que eZ Publish propose un module nommé eZ Personal Page qui permet de fournir à l'utilisateur final une page qu'il peut personnaliser selon ses envies (sorte de iGoogle).

Jahia intègre parfaitement les notions de personnalisation de la restitution, notamment en fonction des droits de l'utilisateur connecté, qui va voir ou non certains modules et certains modes de ces modules. Là encore, le mode 'portail' joue pleinement son rôle.

InfoGlue gère les droits de lecture sur les contenus et les pages. Notons que InfoGlue est nativement connecté à CAS (voir plus loin).

Recueil d'informations

Les sites sont de plus en plus participatifs. Les internautes consultent, mais les internautes contribuent aussi.

Cette nouvelle tendance nécessite des outils permettant de collecter les informations en provenance des internautes : du simple formulaire de commentaire à l'enquête avancée et complète.

Pour répondre à ces attentes, de nombreux CMS incluent des générateurs de formulaires associés à des bases de collecte d'informations. D'autres étendent le principe de « frontend editing » pour faire des internautes des contributeurs à part entière ; et ceci sur l'ensemble des contenus (quels que soient leurs types).

La facilité de recueil d'information est devenue très importante dans les sites de dernières générations ; tout comme l'exploitation de ces données. De ce point de vue, tous les CMS n'offrent pas la même qualité et il est important d'être vigilant sur ce point.

Joomla propose des extensions pour récolter des informations des internautes : chronoforms, artforms, polls, contacts. Cependant ces plugins sont rarement en lien avec le contenu du CMS ; ils sont le plus souvent indépendants.

Spip fournit une interaction assez forte entre les internautes et les administrateurs par l'intermédiaire de formulaire type. Ces formulaires sont activés par l'intermédiaire de balises posées (#FORMULAIRE_SITE, etc.) dans les gabarits (interventions techniques). La création de formulaire à la volée (depuis l'espace de gestion) est possible à travers d'extensions comme forms&tables.

Infoglue n'inclut rien de natif sur ce point.

Typo3 propose de nombreux modules permettant de recueillir des informations : sondage, questionnaire, contenu structuré, ... pas de limite sur ce point.

OpenCMS n'inclut rien de natif sur ce point.

Jahia propose des fonctionnalités de recueil d'informations comme la souscription aux abonnements, les commentaires, etc. Jahia dispose d'un générateur de formulaire (disponible sur la version 5, en attente sur la version 6) permettant de recueillir des informations sous la forme de contenus manipulables.

eZ Publish est particulièrement fort sur ce volet. En effet, lors de la définition de type de contenu, l'administrateur peut paramétrer les attributs qui seront des collecteurs de données (qui se « transformeront » en formulaire). Le principe du frontend editing permet l'édition des contenus par tout type de profil en fonction de la politique de sécurité ; c'est d'ailleurs sur ce principe que fonctionnent les commentaires sous eZ Publish. Le recueil d'informations est également utilisé pour le sondage et les questionnaires ; à noter l'existence de l'extension eZ Survey, outil de questionnaire avancé, qui complète ce point.

Drupal est particulièrement pertinent sur ce point. D'ailleurs, ses nombreux modules Communautaires le démontrent. Son interface unique (front office / back office) facilite le recueil d'informations.

Fonction de portail

Les CMS n'ont théoriquement pas pour objectif de devenir des portails ; c'est-à-dire d'agréger des flux, de réaliser des pages composites, réunissant contenus d'une part, modules applicatifs d'autre part.

Pourtant, lorsqu'on utilise un CMS pour construire un intranet par exemple, on est vite confronté à ce type de besoin.

Dans l'univers des solutions Java, on est bien servi ; des normes existent (Jsr 168 et 286) et l'on peut facilement mettre en œuvre des portlets (blocs) reprenant des flux et interfaces d'applications tierces. Au niveau des CMS PHP, on n'a pas les mêmes standards – on passe souvent par des alternatives comme les iframes et de plus en plus par le javascript (ajax, webservices).

A noter que Smile a développé une solution open source dénommée Esigate. A la fois gestionnaire de cache et agrégateur de contenus, Esigate permet de construire des sites à la volée à partir de sources hétérogènes. Il peut être une vraie solution aux problématiques Portail des solutions Php.

Jahia et Infoglué permettent l'inclusion de portlet (jsr168 pour Infoglué, Jsr 168 et 286 pour Jahia) ce qui donne une dimension Portail à ces produits (sans être réellement des portails comme Liferay par exemple). Ce sont les plus complets sur ce point parmi les CMS même si on n'utilise pas toujours cette dimension.

OpenCMS a beau être basé sur une architecture Java, il ne propose pas de support des spécifications Jsr 168 et 286. Ses possibilités de portail sont très limitées.

Joomla et Spip ne proposent pas de fonctionnalités de portails.

Typo3 et eZ Publish proposent des fonctionnalités de portails à travers le principe de blocs pour Typo3 et de l'extension eZ Flow et eZ Personal Page pour eZ Publish. Ils peuvent également, dans une certaine limite, proposer de la personnalisation (tableau de bord) aux internautes. L'intégration d'applications tierces se fait habituellement par Webservices.

Drupal dispose également de modules permettant la mise en œuvre des pages personnalisés à la iGoogle. Citons par exemple le module Homebox.

Moteur de recherche

On imagine trop souvent qu'un moteur de recherche se justifie principalement par des indexations de documents en texte intégral, ou bien pour rechercher au travers de contenu peu organisé, par exemple dans les messages d'un forum.

C'est un usage restrictif. Beaucoup d'internautes utilisent la fonction de recherche simplement pour éviter de chercher à comprendre l'organisation du site. Ils préfèrent saisir les deux ou trois mots représentatifs de ce qu'ils cherchent et faire 'entrée', puis regarder ce qui leur est proposé, plutôt que de lire le premier menu, réfléchir pour trouver quelle est la rubrique la plus appropriée, parvenir à un niveau inférieur, réfléchir à nouveau, et finalement peut-être faire machine-arrière pour chercher ailleurs.

Ce comportement dans la recherche est inculqué aux internautes par les portails internet, qui sont le plus souvent le point de départ de leur navigation.

Cette démarche directe, qui veut aller droit au but, sans traverser des pages intermédiaires et sans réfléchir, est parfois contraire au souhait du webmaster, qui aimerait que le visiteur prenne son temps, parcoure le plus de pages possibles, lise tout ce qu'on a préparé pour lui ici et là, et en outre, qu'il voie les publicités.

Mais ce que veut le webmaster ce n'est pas ce que veut le visiteur, et il faut se faire à l'idée que le visiteur aura le dernier mot puisque s'il n'est pas au but en trois clics, il ira voir ailleurs.

Il existe trois alternatives pour l'indexation des données : les moteurs de recherche propres aux CMS (développé spécifiquement pour le CMS) ; les moteurs de recherche externe (propriétaire ou open source) qui indexent les sites ; et les moteurs de recherche intégrés au CMS qui se basent sur des solutions reconnues (open source très souvent).

Les moteurs de recherche propres au CMS sont les plus pauvres. Ils se basent sur des découpages et des algorithmes parfois rudimentaires. La pertinence est souvent inexistante ou de mauvaise qualité. Il existe cependant quelques exceptions.

Les moteurs de recherche externes peuvent toujours être intégrés sur un site indépendamment de l'outil CMS retenu, dans la mesure où ils indexent le site en indexant le http, en « crawling ». Ils offrent tous des possibilités d'habillage des pages de résultat par un mécanisme semblable aux gabarits. Certains proposent également des API donnant accès directement aux fonctions de recherche, ce qui permet de mieux les intégrer au sein d'applications spécifiques.

L'intégration de moteurs de recherche externes sera difficile dans le cas où des accès restreints à certains contenus sont mis en place, car d'une part, le moteur doit indexer ces contenus protégés, et d'autre part, le moteur doit vérifier les habilitations de l'internaute pour afficher les résultats de recherche en y incluant ou pas les contenus protégés.

Dans le monde open source, on trouve de très bonnes alternatives comme HtDig, mNoGoSearch, SolR et Lucene. Il s'agit de solutions robustes proposant un périmètre fonctionnel très intéressant.

La meilleure alternative est celle des CMS qui embarquent des moteurs de recherche reconnus (comme Lucene) en totale liaison avec la politique de sécurité. C'est une tendance qui se développe tant au niveau des CMS que des portails.

Cet interfaçage est un bon point techniquement. Il ne faut cependant pas oublier de juger un moteur de recherche sur ses fonctionnalités : recherche dans des documents (PDF, word, ...), recherche par rubrique, gestion de la pertinence, prise en compte des pluriels, recherche de proximités sémantiques, etc.

Spip offre des fonctions de recherche intégrées mais très simplifiées. C'est une recherche très sommaire, qui ne traite pas les opérateurs logiques, l'indexation des documents, et le classement par pertinence. Pour un besoin sérieux, notre recommandation est de recourir à un moteur externe.

Joomla permet des recherches selon 3 modes : « AND », « OR », et « expression exacte ». Les résultats peuvent être triés suivant la date de publication (plus récent/plus ancien), la popularité, l'ordre alphabétique, la section/catégorie et le type de contenu.

Drupal offre une recherche avec opérateurs logiques et permet de filtrer les résultats selon leur catégorie et leur type de contenu. Grâce au module « Search Attachements », Drupal peut également effectuer des recherches full-text dans les documents attachés. Notons qu'un module « Apache Solr Search Integration » permet de coupler Drupal au moteur Open Source Solr.

eZ Publish offre une fonction de recherche plus avancée, avec notamment la possibilité de rechercher par champ de contenu (titre=...), mais aussi une recherche plein-texte sur les contenus et les documents joints. eZ Publish propose désormais nativement l'extension eZ Find qui est un couplage entre eZ Publish et le moteur de recherche open source Solr. Il s'agit d'un compromis puissant qui offre un périmètre très large (recherche par facettes, etc.) tout en respectant la sécurité des contenus.

Typo3 intègre un moteur de recherche qui indexe les contenus, y compris les documents joints, et la liste des résultats est triée sur un critère de pertinence calculé. Ce module reste inférieur à htDig ou mNoGoSearch, mais il est très performant et propose des résultats organisés par dossier. Pour des besoins avancés, un moteur de recherche externe restera nécessaire.

Jahia intègre le moteur Lucene issu du projet Apache, qui permet d'indexer l'ensemble des contenus du site, sans utiliser le http mais avec des connecteurs spécifiques pour la lecture des fichiers et de la base de données.

InfoGlue intègre le moteur de recherche Lucene, qui parcourt les pages du site et les indexe, proposant ainsi une recherche full-text sur l'ensemble des contenus. Malheureusement, cette recherche est uniquement disponible sur le back-office. Pour le front office, il est nécessaire de mettre en place un moteur de recherche externe (comme Solr) en le connectant à partir des « intercepteurs ».

OpenCMS intègre également le moteur de recherche Lucene. Le contenu est indexé dès la publication d'un élément. L'administrateur dispose d'une interface de configuration et de suivi de l'indexation.

Droits et sécurité d'accès

Un bon CMS doit offrir une gestion souple des habilitations et droits d'accès. Pour bien faire, cette sécurisation doit être transparente aux yeux des contributeurs.

Utilisateurs, groupes et rôles

Les sites de petite et moyenne envergure ont souvent un contributeur unique, le webmaster. Qu'il rédige lui-même les contenus, ou plus couramment, qu'il les collecte au sein de l'entreprise, il est du moins le seul à introduire des contenus dans le site. Il s'ensuit bien sûr qu'il est habilité à saisir ou modifier les contenus en tout point du site.

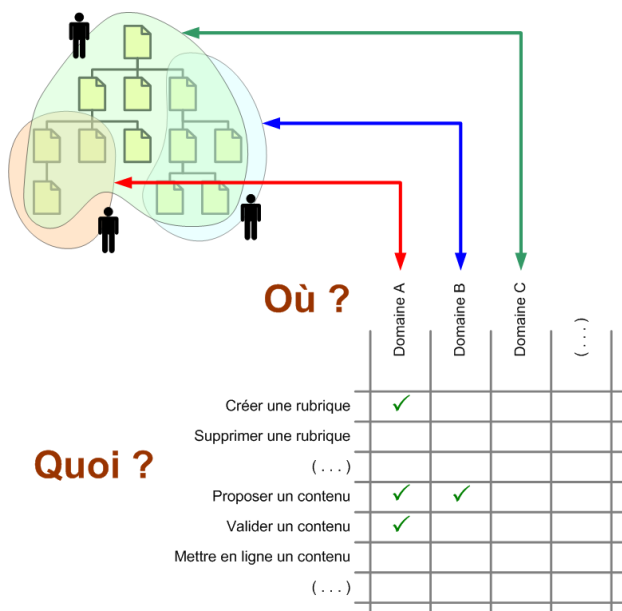
Sur des sites de grande envergure, il n'est plus possible qu'une seule personne ait la charge de saisir tous les contenus. De plus lorsque les contenus sont en fait rédigés en dehors du CMS, ils doivent être ressaisis, ou bien copiés/collés, dans le CMS par le webmaster, ce qui représente une étape inutile.

Pour toutes ces raisons, **il peut être utile de faire directement interagir les contributeurs premiers sur le CMS, en leur offrant une interface pour la saisie, puis la modification, de leurs articles.** De plus, dans le contexte d'un Intranet, la contribution des collaborateurs est un facteur de participation, et donc d'adhésion : l'Intranet n'est plus un outil de communication 'descendante', mais devient un outil d'échange.

Dans ce contexte, il sera le plus souvent nécessaire d'accorder aux contributeurs des **droits limités à certaines parties du site.** Ces droits sont bien sûr définis en référence à l'organisation hiérarchique des contenus.

On définit ainsi par exemple une rubrique 'Ressources Humaines', qui peut être le point de départ d'une importante arborescence de contenus, et l'on accorde à certains contributeurs le droit de saisir des articles relevant des Ressources Humaines. Les mêmes contributeurs ne pourront pas saisir des articles relevant de la rubrique 'Produits et Services'.

Les droits de chacun sont donc définis *en référence à l'organisation des contenus*. Mais doivent également être définis *en référence aux actions possibles sur ces contenus*.



Ainsi tel intervenant pourra *proposer un article*, mais ne pourra pas le valider ou le mettre en ligne. Tel autre pourra valider un article dans les rubriques dont il est en charge, mais ne sera pas autorisé à créer de nouvelles rubriques, etc.

Les CMS doivent donc offrir une gestion d'habilitation à deux axes : le

premier est l'axe des contenus, et de leur organisation, le second l'axe des fonctions liées à la gestion de ces contenus : consulter, mettre à jour, valider, publier, etc...

En général, les droits se définissent au niveau de l'utilisateur ou au niveau d'un groupe d'utilisateurs ; avec un cumul des droits lorsqu'un utilisateur appartient à plusieurs groupes. Pour faciliter la configuration des habilitations, la majorité des CMS utilisent la notion de « Rôle » ; il s'agit d'un regroupement type de droits qui peut être associé à un utilisateur ou à un groupe.

La définition de la politique de sécurité d'un site est une étape importante qui doit être préparée en amont de la réalisation, dès la conception. Pour faciliter cette mise en place et assurer un bon suivi, de nombreuses fonctionnalités ont été ajoutées par les éditeurs et les communautés. Certains CMS proposent la délégation de droits, d'autres mettent en œuvre des héritages de droits, etc. ; tout est fait pour simplifier la gestion du site.

Spip permet à plusieurs contributeurs d'agir sur le site avec des rôles prédéfinis. Pour Spip, il s'agit d'Auteurs, qui sont soit rédacteurs soit administrateurs, et qui accèdent à tout ou partie du site. Un rédacteur ne peut que créer un contenu ; un administrateur peut créer et valider un contenu pour une rubrique donnée (seuls les comptes administrateurs peuvent être restreints sur une rubrique donnée – pas deux). La gestion des droits sous Spip est très limitée mais correspond à un fonctionnement classique qui peut satisfaire de nombreux utilisateurs. Pour étendre le fonctionnement natif, des extensions existent.

Joomla propose 3 grand niveaux de contribution : « author », « editor » et « publisher » qui permettent respectivement de modifier, créer et publier du contenu. Chaque article peut être associé à un auteur donné par un utilisateur disposant de droits supérieurs. Dans le cadre d'un simple site web, ce système est amplement suffisant mais aussi très vite dépassé dans le reste des cas. La version 1.6 doit apporter une gestion des ACL (Access Control List).

Drupal dispose d'une gestion très fine des droits. Il est possible de créer un nombre quelconque de rôles et de leur affecter des droits pour chaque fonctionnalité de chaque module. Ce modèle permet à Drupal de s'adapter à n'importe quel besoin. Pour restreindre, les accès à différents contenus, plusieurs modules existent : « Node privacy by role », « Taxonomy access », etc.

Typo3 possède une gestion des droits des contributeurs bien plus complète que celle de Spip. Comme OpenCms, elle est inspirée du modèle Unix, avec les notions de *user*, *group*, *all*. La plus petite entité est l'attribut (champs) d'un contenu. On peut aussi paramétrer les droits au niveau des colonnes, des langues, des blocs, des arborescences, etc. Typo3 propose de l'héritage de groupes.

OpenCms propose une gestion des droits inspirée de Unix : chaque section de l'arborescence est vue comme un répertoire, et les droits de *lire*, *écrire*, *voir* sont attribués par rapport aux périmètres croissants : *user*, *group*, *other*, à savoir celui qui a écrit l'article, ceux de son groupe, et les autres.

eZ Publish permet de définir des rôles, des groupes et des utilisateurs. Les rôles comprennent l'ensemble des droits autorisés, pour chaque module, pour chaque fonctionnalité et pour un périmètre donné (langue, rubrique, section, état, etc.). Ce même système permet de restreindre les droits en consultation sur le site, ou en écriture sur un forum par exemple. Par contre, on ne peut voir par contenu les droits affectés.

Gestion de contenus – les solutions open source

Jahia permet de définir les profils et les droits nécessaires à une gestion décentralisée, avec les notions de groupes, utilisateurs, manipulables dans tous les sens. La gestion des droits est aussi appliquée aux modules du portail, ce qui permet d'avoir une identification unique pour l'ensemble des applications mises en œuvre. Contrairement à eZ Publish, la gestion des droits se configure au niveau de chaque contenu avec, à noter, une notion d'héritage très intéressante. Par contre, il n'existe pas de notion de groupe de groupe (hiérarchie).

InfoGlue permet de définir des groupes et des rôles, associés aux utilisateurs, qui peuvent donc accéder à certains modules, certains contenus et certaines actions uniquement.

Annuaire

Une majorité d'entreprises dispose désormais d'un annuaire central. Il s'agit d'un référentiel regroupant l'ensemble des collaborateurs de la société (avec leurs fiches descriptives).

Généralement, cet annuaire est utilisé pour stocker les identifiants et les mots de passe des utilisateurs. Dans certains cas, les habilitations y sont directement enregistrées ; même si la plupart du temps, cela se limite à la notion de groupe (groupe « Ressources Humaines »).

Les CMS proposent désormais un interfaçage avec ces annuaires (OpenLDAP, AD, etc.). Cela permet de centraliser les informations des utilisateurs – le référentiel c'est alors l'annuaire LDAP ; pas le CMS. C'est essentiel pour garantir une bonne intégration du CMS dans le système d'informations de l'entreprise.

De plus, les utilisateurs n'ont plus à mémoriser un nouveau mot de passe – ils utilisent le même que celui permettant de se connecter à leur machine.

Il existe différentes mises en œuvre possibles :

- L'annuaire peut être utilisé pour initialiser la base Utilisateurs du CMS (avec éventuellement une synchronisation régulière).
- L'annuaire peut être utilisé à chaque identification (la base Utilisateurs du CMS n'est alors plus utilisée).
- L'annuaire peut être utilisé pour chaque identification tout en mettant à jour systématiquement les données du profil (contenu dans la base du CMS) pour être utilisées durant toute la session de l'utilisateur (hormis le mot de passe).

Certains algorithmes de connexion permettent de récupérer les groupes LDAP de l'utilisateur et d'effectuer le mapping avec les groupes du CMS. C'est une façon, assez simple, de gérer les habilitations depuis l'annuaire d'entreprise.

Enfin, notons que certains CMS autorisent l'utilisation conjointe de la base locale d'utilisateurs et de l'annuaire distant.

Drupal permet de se connecter à un annuaire LDAP. Pour cela il faut activer l'extension `ldap_integration`. Pour utiliser uniquement l'authentification, il faut activer le plugin `ldapauth`. Il est possible de faire un mapping entre les rôles de Drupal et les groupes LDAP grâce au module `ldapgroups`.

Depuis la version 1.5, **Joomla** gère l'identification via LDAP. Sur les versions antérieures, il était nécessaire d'utiliser une extension comme `JAAuthTools` par exemple.

eZ Publish permet de s'identifier depuis un annuaire LDAP. Un script de synchronisation est également disponible pour mettre à jour les utilisateurs enregistrés sous eZ Publish.

Les utilisateurs de **Typo3** peuvent s'authentifier via un annuaire LDAP par l'intermédiaire d'une extension (`ig_ldap_sso_auth` par exemple). Elle est principalement destinée aux serveurs OpenLDAP et eDirectory. Comme eZ Publish, elle permet la synchronisation des données utilisateurs. Typo3 gère également OpenID.

Infoglupe gère très bien l'identification via LDAP. Cela nécessite l'activation de modules dédiés mais permet l'identification et l'autorisation d'accès en rapprochant les utilisateurs, groupes et rôles des éléments de l'annuaire. La structure « particulière » proposée par le connecteur natif d'Infoglupe oblige parfois à écrire un autre connecteur.

OpenCMS peut gérer le support LDAP mais en passant par une extension payante de l'éditeur : Alkacon OCEE LDAP Connector.

Jahia supporte l'identification via un annuaire LDAP (Active Directory, eDirectory, LemonLDAP, etc.) et permet la synchronisation de certains attributs.

Spip supporte nativement l'identification via un annuaire LDAP. La gestion des autorisations peut être mise en place même si elle est limitée à la gestion des droits des utilisateurs (1 rubrique).

Identification unique

L'identification unique (SSO) permet d'accéder à un ensemble d'applications Web sans devoir saisir son identifiant et mot de passe à plusieurs reprises.

L'identification unique est particulièrement adaptée aux intranets où l'authentification sur un poste Windows, par exemple, permet d'accéder aux applications qui composent l'intranet sans avoir à passer par des formulaires d'identification supplémentaires.

Le SSO peut être mis en œuvre au travers de différentes solutions. Citons par exemple CAS (Central Authentication Service), Kerberos, etc. Pour mettre en place une identification unique, le CMS doit être en mesure de supporter ces solutions ; pour CAS, on dit que les applications doivent être CASsifiées.

Les CMS qui ne supportent pas nativement le SSO peuvent être modifiés ; mais cela nécessite une intervention délicate mise en œuvre par des experts techniques (Smile intervient régulièrement sur cette problématique).

Pour beaucoup de CMS, le SSO est mis en œuvre à travers de plugins. Soyez vigilant sur la qualité de ces derniers.

Plusieurs implémentations SSO ont déjà été réalisées sur **eZ Publish** (par Smile notamment) mais il n'existe pas de support officiel de l'éditeur. Par contre, eZ Publish propose des « SSO Handler » qui permettent la mise en œuvre du Single Sign On comme CAS, Kerberos, etc. mais cela nécessite une intervention en développement.

Joomla ne propose aucun support officiellement. On trouve cependant une ou deux extensions permettant la mise en place du SSO mais sans aucune garantie de fiabilité et de sécurité.

Spip supporte nativement LDAP ce qui est un bon début. Concernant le SSO, une extension relativement complète était disponible avant la version 2.x. Ce plugin n'est plus à jour et n'offre aucune garantie.

Drupal intègre la gestion SSO par l'intermédiaire de plugin. On trouve notamment une bonne implémentation de CAS.

Infoglupe supporte nativement l'implémentation du SSO ce qui est relativement rare dans l'univers des CMS ; un point important à souligner.

OpenCms n'inclut pas nativement de support du SSO. L'implémentation est possible et des tutoriaux de qualité existent dans la communauté mais cela nécessite une intervention en développement et idéalement le support de LDAP (qui est payant).

Jahia supporte nativement la plupart des systèmes d'authentification unique : CAS, NTML, etc. Pour les protocoles qui ne sont pas encore supportés, Jahia propose un système de « plugin d'authentification » qui permet d'étendre le périmètre de base à d'autres protocoles.

Typo3 dispose d'extensions de qualité permettant la mise en œuvre du SSO (LDAP/SSO Authentification par exemple). Cette extension est communautaire et doit être vérifiée lors de son implémentation par des prestataires comme Smile.

Socle technique

Nous sommes de plus en plus nombreux sur internet. Les volumes de consultation des sites explosent. Le moindre blog peut vite recevoir des milliers de visites par mois, voire par jour.

Cette dimension ne peut plus être occultée. Au delà du périmètre fonctionnel d'un outil, de la facilité de prise en main, du dynamisme de sa communauté, il faut absolument tenir compte de sa capacité de montée en charge. Quelle déception si le travail de nombreuses semaines de conception ne pouvait être vu par tous à cause d'une architecture défaillante !

Dans cette rubrique, nous vous proposons une approche des socles techniques ; si vous souhaitez aller plus loin, n'hésitez pas à lire notre livre blanc sur les architectures hautes performances.

Framework

La qualité, la performance et la stabilité des frameworks et bibliothèques de développements Open Source n'est plus à prouver. Elles ont fait la réussite de nombreuses applications parmi les plus utilisées au monde.

Aujourd'hui, ces mêmes outils sont régulièrement utilisés dans la réalisation des outils de gestion de contenu. C'est particulièrement vrai dans le monde JAVA même si l'univers PHP n'a plus rien à envier avec des frameworks comme Zend, Symfony ou CakePHP de très grande qualité.

Un framework permet de réduire les temps de développement des applications en répondant de façon efficace aux problèmes les plus courants rencontrés par les développeurs (comme une couche d'abstraction pour travailler sur un ensemble de bases de données). Le temps ainsi gagné sur les questions génériques peut être mis à profit pour les parties spécifiques de l'application, les fonctionnalités clés du CMS dans notre cas.

De plus, l'utilisation d'un même framework contribue à l'industrialisation du processus de développement : la structure de l'application et les méthodes de programmation sont standardisées, la maintenabilité est d'autant meilleure.

Un framework est le fruit du travail de dizaines (voir de centaines) de personnes qui s'appliquent à corriger les problèmes ou les failles de sécurités découvertes par l'ensemble des utilisateurs et à proposer de nouvelles fonctionnalités. C'est pourquoi les bibliothèques d'un framework sont en général mieux conçues et mieux codées, mais aussi mieux testées et donc plus robustes, que ce que ne pourrait produire un unique programmeur (ou une petite équipe). Outre le gain de temps, on obtient un important gain en termes de qualité.

Les CMS basées sur un framework offrent, généralement, une forte extensibilité, dans la mesure où les composants du framework sont parfaitement documentés et donc ouverts.

De nombreux CMS ne reposent pas sur un framework ; ce n'est pas une fin en soi mais les sources doivent alors être étudiées en détail pour garantir la capacité de montée en charge, de stabilité et de sécurité.

Spip, Joomla et Drupal ne se basent pas sur un framework. Il s'agit d'applications « from scratch ».

eZ Publish se base en partie sur son framework eZ Component. Petit à petit toute l'application devrait être réécrite sur la base de ce framework.

Typo3 ne se base pas sur un framework. Par contre, la prochaine version, la 5, devrait être entièrement réécrite avec le framework maison Flow3.

Jahia se base sur un ensemble de briques java open source comme hibernate, struts, spring, gwt, jgroups, pluto, etc.

OpenCMS s'appuie sur un ensemble de briques « java » : Lucene, Dom4J, JTidy, Doomdark JUG, Log4j, Digester, Commons DBCP/Pool, Quartz, Junit, etc.

Infoglué se base, entre autres, sur : Webwork1 pour les vues du backoffice, Castor pour la persistance des données CMS, Hibernate 2 pour la persistance des données de workflow OS, OpenSymphony workflow pour la gestion des workflow spécifique, Velocity pour la définition des vues du backoffice.

API distante

Une API est une interface de programmation (Application Programming Interface) constituée d'un ensemble de fonctions, procédures ou classes mises à disposition des applications pour garantir son interopérabilité.

Généralement les CMS sont contrôlés, pilotés depuis une interface d'administration (back-office). Pourtant, il peut être utile d'interagir à distance, ou via un programme tiers, avec le CMS.

C'est le cas notamment lorsqu'on dispose d'un Système d'Information complet qui gère des fiches produits. Ces fiches peuvent être mises à jour depuis le SI et publiées sur le CMS à travers un script informatique (de tout genre). Ce script utilise généralement l'API du CMS ; dans le cas contraire, le script doit travailler directement avec la base de données du CMS ce qui ne garantit pas son intégrité.

Pour les projets d'envergure, l'interfaçage avec des outils tiers est incontournable ; dans ce cas, l'utilisation d'une API fonctionnelle et bien documentée est très importante.

Typo3 intègre SOAP et propose une API complète.

Jahia propose des éléments de type Rest. L'implémentation complète de JackRabbit devrait permettre, à terme, d'atteindre l'ensemble des contenus de cette manière.

OpenCMS ne supporte pas nativement de protocole d'échange type SOAP. Par contre, la communauté a commencé l'écriture de module permettant l'échange sur la base du protocole JAX-WS.

Infoglue dispose nativement d'une API complète accessible via SOAP.

eZ Publish propose un serveur SOAP avec quelques fonctions. Il est possible de les étendre avec l'API « local » d'eZ Publish.

Joomla dispose de quelques extensions (comme XML_RPC_Joomla) permettant l'échange via XML-RPC.

Drupal dispose également d'une extension permettant l'échange suivant le protocole XML-RPC.

Spip ne supporte pas de protocole d'échange.

Plugins et extensibilité

Les CMS ne peuvent pas tout faire – ou du moins nativement. Ils offrent déjà un scope fonctionnel très intéressant qui couvre la plupart des besoins.

Mais le périmètre des uns, n'est pas le périmètre des autres. Il faut donc pouvoir étendre le CMS sans avoir à modifier son noyau, c'est-à-dire le cœur du logiciel.

Cette notion d'extensibilité est désormais proposée par la majorité des CMS ; avec plus ou moins de qualité (c'est d'ailleurs un point d'adhésion pour les milliers de développeurs d'une communauté).

Pour certains CMS, il existe des centaines voire des milliers d'extensions. Dans ce cas, il faut être capable de faire la part des choses et de ne retenir que les meilleures. Les commentaires de la communauté sont très intéressants pour ça.

Pour les CMS s'orientant vers ce modèle (noyau + extension), la différence se fera essentiellement sur la capacité du socle à proposer un tout opérationnel et cohérent (accès aux fonctionnalités, dépendance, etc.).

Si vous ne trouvez pas « chaussure à votre pied », vous pouvez développer une nouvelle extension. Selon le CMS retenu, cela peut être très simple comme très compliqué.

Joomla permet d'ajouter des fonctionnalités au travers de l'ajout de module. Le site <http://extensions.joomla.org/> référence les extensions disponibles (plus de 4 000). Certaines sont essentielles pour compléter le périmètre parfois limité de Joomla.

Infoglué n'offre pas de système d'extensions comme on peut le voir par ailleurs ; le moteur d'Infoglué possède plusieurs points particuliers appelés points d'interceptions (sorte de surcharge de comportement natif). Son architecture Java lui permet par contre d'incorporer des portlets.

Spip a introduit la notion de plugins depuis sa version 1.9. Sa communauté active apporte de bons plugins bien référencés. Spip propose d'ailleurs un site dédié à ses extensions (<http://www.spip-contrib.net>) qui présente les compatibilités et les notations des utilisateurs. L'activation des plugins est relativement simple même si les fichiers des extensions doivent être déposés préalablement sur le serveur.

eZ Publish dispose principalement d'une communauté de professionnel ce qui explique le moindre nombre d'extensions (plusieurs centaines tout de même). Celles-ci sont cependant de grande qualité et touchent souvent des sujets de hauts niveaux.

Typo3 peut facilement être étendu au travers de ces milliers de plugins. Le site <http://typo3.org/extensions/> les recense tous sans exception ! On peut facilement consulter les extensions les plus populaires comme tt_news ou powermail.

OpenCMS propose quelques extensions sur son site principal (<http://www.opencms.org/en/download/contributions.html>). Elles ne sont pas très nombreuses mais de bonnes qualités et surtout très utiles (KonaKart Shopping Engine, jBPM Workflow Engine, etc.).

Drupal propose plus de 10 000 extensions ! Il n'y a pas un module que vous cherchiez qui n'existe pas. Vous pouvez consulter les extensions disponibles sur l'adresse <http://drupal.org/project/Modules>. Sur la page « view module usage statistics » vous pouvez observer les plugins les plus utilisés : views, cck, token, etc. L'installation de modules est simple et s'appuie sur des règles de dépendance.

Jahia propose un site communautaire, <http://www.jahia.org>, mais il ne propose pas d'extensions à déployer facilement. Comme Infoglué, son architecture lui permet d'accueillir des portlets.

Performance

Comme évoqué en introduction de la rubrique « Socle technique », la performance des CMS est devenue un concept fondamental de la gestion de contenu.

Impossible de construire un site ambitieux sans prendre en compte cette dimension.

Pourtant, il s'agit là du point le plus complexe à évaluer. La performance dépend du contexte de réalisation (quels modules ? quelles options ? quelles personnalisations ?) et du contexte d'exploitation. Il n'existe aucune échelle de performance hormis peut être celles d'intégrateurs comme Smile, qui reste cependant à contextualiser avec votre besoin.

Les références utilisant le CMS peuvent être une bonne indication (à architecture équivalente) ; tout comme la réalisation de mesure de tests des différents outils dans des conditions similaires (quel nombre de sessions simultanées peuvent être servies en maintenant des temps de réponse inférieurs à 1 seconde pour 90% des cas ?).

Au-delà des tests, un CMS performant est un CMS qui propose un bon système de cache, c'est-à-dire gardant en mémoire les pages plus demandées tout en les rafraîchissant lors de mises à jour. Certains vont jusqu'à proposer une version statique (Html) du site, voir un interfaçage complet avec des solutions comme Squid ou Varnish (« proxy-cache » ou « reverse-proxy »).

Enfin, notons que certains CMS proposent des architectures extensibles, c'est-à-dire des architectures qui peuvent évoluer (nombre de serveur par exemple) selon la charge à tenir. Nous vous invitons à lire notre livre blanc sur les plateformes web hautes performances pour en savoir plus sur cette question.

Typo3 dispose de nombreux éléments de cache, par page et par bloc. Ce cache peut être géré de différentes manières : par fichier, par base et par memcached. Il est également possible d'installer Typo3 en mode cluster et de proposer une génération statique des pages.

OpenCMS offre de bonnes performances.

Jahia propose d'excellentes performances comme souvent avec les architectures JEE. Il propose nativement un mode cluster et du cache fragmentaire. Des benchmarks sont disponibles sur le site de l'éditeur.

Drupal offre d'excellentes performances en mode déconnecté grâce, notamment, à un système de cache très performant.

Infoglupe fait partie des CMS les plus « costaud » du marché.

eZ Publish offre de bonnes performances en utilisant correctement les nombreux systèmes de cache fournis. Pour obtenir de meilleures performances, eZ Publish permet la génération statique des pages Html et l'utilisation de reverse-proxy comme Squid ou Varnish (avec un lien de rafraichissement). Pour les architectures hautes-performances, eZ Publish propose des montages en cluster.

Spip offre de très bonnes performances avec son système de cache natif. Sa configuration est relativement sommaire mais permet de répondre à la mise en œuvre d'un site de contenu sans personnalisation.

Joomla offre des performances acceptables qui peuvent cependant être vite dépassées lors de fortes consultations.

SYNTHÈSE

Avant de conclure, soulignons que notre analyse était construite autour des *concepts fondamentaux* de la gestion de contenu, et non autour des *critères de choix* (qui sont plus nombreux encore et que vous pouvez retrouver dans notre livre blanc : 200 questions pour choisir un CMS). Notre objectif n'était pas ici de construire un tableau croisé évaluant chacun des CMS du panel sur chacun des critères. Notre objectif, comme nous l'avons indiqué en préambule, était de proposer notre vision de la problématique de gestion de contenu, et de positionner les meilleurs outils.

Les critères de choix, quant à eux, se réfèrent pour partie à ces concepts fondamentaux, mais incluent aussi d'autres facteurs : dynamisme de la communauté, références, documentation, etc.

Il n'y a pas un outil qui ressortirait comme le gagnant de l'étude : la sélection s'est faite en amont, c'est à dire que nous n'avons présenté ici que des outils de qualité, qui sont parmi les meilleurs choix à ce jour pour construire votre site, ou bien gérer plus largement vos contenus.

Les solutions open source sont à l'informatique ce que les médicaments génériques sont à la pharmacie : une fois qu'un domaine est devenu mature, que les concepts sont stabilisés, les bonnes pratiques et les solutions relèvent d'un patrimoine commun. C'est progressivement ce qui est en train de se passer dans le domaine de la gestion de contenu. Sur le bas de gamme, la cause est entendue et il n'y a pratiquement plus d'offre autre que open source. Sur le haut de gamme, certains produits offrent une alternative plus que crédible aux offres propriétaires.

Smile a mis en œuvre des solutions de gestion de contenu sur plusieurs centaines de sites, en Internet ou Intranet, et peut mettre son expertise au service de vos projets.