

Description du processus de choix d'un CMS pour le projet PLUME¹

Unité Réseaux du CNRS (UREC / CNRS)

Pierre-Yves GOSSET

pierre-yves.gosset@urec.cnrs.fr

v0.4 - 19/06/2007

¹ <http://www.urec.cnrs.fr/plume>

Table des matières

Description du processus de choix d'un CMS pour le projet PLUME.....	1
1 -Introduction.....	3
2 -Résumé de la démarche.....	3
3 -Analyse du besoin.....	4
3.1 -Présentation générale.....	4
3.2 -Gestion des droits.....	4
3.3 -Processus de Workflow.....	5
3.4 -Outils collaboratifs.....	5
3.5 -Manipulation de données.....	5
3.6 -Autres fonctionnalités.....	5
4 -Contraintes.....	5
5 -Méthodologie.....	6
6 -De la difficulté d'une première sélection : le choix.....	8
6.1 -Critères de sélection.....	8
6.2 -Note sur le cas de HAL.....	9
7 -Seconde sélection : tests.....	10
8 -Catégorisation et classement.....	11
9 -Troisième sélection : adéquation avec le projet PLUME.....	13
10 -Choix final.....	13
11 -Conclusion.....	15
Annexes.....	16
1 -Glossaire et définitions.....	16
2 -Résultats de la seconde évaluation.....	17
2.1 -Drupal.....	17
2.2 -e107.....	19
2.3 -eZpublish.....	20
2.4 -Jaws.....	21
2.5 -Joomla.....	22
2.6 -Lodel.....	24
2.7 - MODx	25
2.8 - MySource	26
2.9 -SPIP.....	27
2.10 -Textpattern.....	29
2.11 -Tiki CMS.....	30
2.12 -Typo3.....	31
2.13 -Xaraya.....	32
2.14 -Xoops.....	34

1 - Introduction

Le projet PLUME² vise à Promouvoir les Logiciels Utiles, Maîtrisés et Économiques pour la communauté de l'Enseignement Supérieur et de la Recherche. Un des objectifs techniques est de mettre en production un serveur Web de référence de fiches descriptives de logiciels Utiles Maîtrisés et Économiques.

Dans le cadre de ce projet, l'équipe en charge de sa réalisation a dû évaluer les différentes plateformes techniques qui seraient à même d'accueillir les fiches présentant les logiciels.

Le but de ce document est de décrire les différentes étapes et méthodes d'évaluation qui nous ont mené au choix du système de gestion de contenus finalement retenu. Il faut souligner que l'outil choisi correspond à des besoins spécifiques et il ne faut pas interpréter le choix comme générique.

2 - Résumé de la démarche

Sur le serveur PLUME les logiciels sont présentés sous forme de fiches rédigées, relues et validées par des membres de la communauté. Le site propose aussi d'autres contenus et outils, comme des informations sur les événements concernant la communauté (actualités, séminaires, formations, etc) ou des outils de collaboration en ligne permettant aux membres de partager leurs retours d'expériences sur l'utilisation et leurs pratiques des logiciels.

Afin de pouvoir publier et structurer ce contenu rédigé par des auteurs multiples, PLUME repose sur un système de management de contenus (CMS, Content Management System).

Une première maquette de PLUME, reposant sur le CMS SPIP, a été mise en place fin 2006. Elle a permis de mieux identifier les écueils à éviter et les besoins difficilement discernables sans confrontation à la pratique. Après avoir rédigé une "Expression des besoins et des fonctionnalités demandées" plus précise, l'équipe en charge de PLUME a donc décidé de chercher le CMS qui serait le plus adaptable et cohérent par rapport au projet.

L'offre d'applications de type CMS représentant plusieurs centaines de logiciels, cette sélection s'est faite en 4 phases distinctes, étalées sur une période d'environ 3 semaines.

D'abord, ont été éliminés de la liste tous les logiciels ne correspondant pas aux "contraintes fortes" du projet PLUME : contrainte temporelle (mise en exploitation de l'outil en moins de 3 mois), contrainte technique due aux compétences internes (l'application doit fonctionner en environnement Linux, Apache, Php, MySQL), contrainte humaine (ressources internes limitées), etc.

Cette première liste de critères établie, l'évaluation a pu débuter sur une sélection de 14 applications (Drupal, e107, eZpublish, Jaws, Joomla, Lodel, MODx, MySource, SPIP, Textpattern, Tiki CMS, Typo3, Xaraya et Xoops).

Ensuite, ces outils ont fait l'objet d'une phase de tests relativement rapides (installation de l'application sur un serveur de tests, puis quelques heures d'étude du fonctionnement) afin d'en dégager les principales caractéristiques. Les critères portaient notamment sur la présence d'une documentation de qualité, une évaluation de la courbe d'apprentissage du produit, mais aussi et surtout sa flexibilité, sa modularité et sa pérennité.

Les applications ont alors été réparties dans différentes catégories "d'usages" susceptibles d'influer sur leurs modalités d'utilisation par les différents acteurs du projet (visiteur, contributeurs, responsables de rubrique, rédacteur en chef, développeurs, etc).

2 <http://www.urec.cnrs.fr/plume>

A l'issue de cette sélection, 5 applications semblaient être des candidates potentielles intéressantes pour le projet.

Dans une troisième phase, ces 5 CMS - les meilleurs de chaque catégorie - ont fait l'objet de tests plus poussés (quelques jours). Les éléments évalués permettaient de dégager les possibilités d'adéquation entre les applications et le projet PLUME.

Ainsi rentraient en compte les critères de sécurité, performances, extensibilité, simplicité d'utilisation/administration ou pérennité des applications.;

Au terme de cette troisième phase, deux CMS (Drupal et SPIP) restaient à départager

Pour cette dernière phase de sélection, les deux applications - d'une conception et d'une logique très différentes - ont donc fait l'objet d'une configuration aussi proche que possible de celle souhaitée lors de l'expression fonctionnelle des besoins. Les 2 "maquettes" ont alors été comparées sur une liste de plus de trente critères pondérés par priorités.

Au final, le CMS Drupal nous a paru le plus adapté dans le cadre du projet PLUME, notamment grâce à sa très grande flexibilité de configuration, sa base d'utilisateurs importante et garante d'une certaine pérennité ou ses nombreux modules additionnels disponibles (dont certains répondent parfaitement à des contraintes spécifiques au projet PLUME, comme la gestion fine du workflow de publication).

3 - Analyse du besoin

Préalablement ou en parallèle de la rédaction des spécifications fonctionnelles de la plate-forme, un certain nombre de besoins ont été identifiés.

On pourra noter que cette analyse de besoins aura été grandement facilitée par la mise en place d'une maquette du projet PLUME, permettant de mieux cerner les difficultés et les leviers du projet.

Sans reprendre ici l'intégralité des spécifications fonctionnelles, nous pouvons en dégager les points les plus importants ou spécifiques au projet PLUME.

3.1 - Présentation générale

La plate-forme prévue doit permettre aux membres du projet PLUME un travail collaboratif, en particulier la rédaction des fiches et leur relecture selon les processus mis en place par le projet. Ceci implique la gestion d'un workflow. La publication des fiches ainsi élaborées sur le site de publication doit être facile. Les fiches finalisées seront accessible à tous mais une partie du site sera réservée aux lecteurs de la communauté Enseignement Supérieur et Recherche. Enfin, l'objectif est une première version accessible aux visiteurs courant septembre 2007 et une version finalisée pour la fin 2007, sachant que les ressources humaines disponibles sont d'une personne à plein temps jusqu'à la fin 2007. Cette contrainte est fondamentale pour la compréhension des choix.

3.2 - Gestion des droits

Certains éléments du site seront en accès restreint. Cela peut concerner des parties complètes de PLUME (par exemple certaines rubriques) mais aussi certaines fonctionnalités, par exemple la publication de commentaires associés aux fiches doit pouvoir être contrôlée (en terme d'accès en lecture, écriture, ou modification).

Les droits de publication doivent pouvoir être gérés par rôles. Ainsi, un visiteur non identifié ne pourra pas soumettre de fiche PLUME, un visiteur enregistré et connu pourra proposer des fiches, mais pas les rédiger avant accord d'un responsable de rubrique, etc.

3.3 - Processus de Workflow

Le processus de workflow de publication est spécifique à PLUME, et l'outil retenu doit pouvoir s'y adapter. Ainsi, les principales étapes sont : proposition, rédaction, relecture, validation, publication (chaque étape étant composée de différentes sous-étapes).

Les spécificités sont notamment :

- un workflow non linéaire : en cas de retard ou d'abandon d'une fiche, il est possible de revenir à une étape précédente du workflow. Il en est de même à certains moments de la "vie" d'une fiche (qui sera par exemple automatiquement déclarée obsolète au bout de 6 mois sans mise à jour).
- un processus de relecture rigoureux : les fiches sont relues par deux personnes, qui peuvent modifier/compléter/commenter la fiche. Mais ces informations ne sont pas directement transmises à l'auteur, elles seront compilées/validées par un responsable de rubrique, qui soumettra ce travail au rédacteur initial pour validation.
- la possibilité pour l'auteur de "dépublier" à tout moment sa fiche.
- la possibilité de gérer des actions (envoi de mails, notamment) à certaines étapes du workflow.

3.4 - Outils collaboratifs

Un des objectifs du projet est de fédérer la communauté des utilisateurs de logiciels dans la communauté de l'Enseignement supérieur et de la Recherche. Pour cela un certain nombre d'outils doivent être mis à leur disposition pour leur permettre d'échanger et de collaborer de la façon la plus efficace possible : commentaires liés aux fiches, forums spécialisés (à accès restreint ou non), wiki, etc.

3.5 - Manipulation de données

L'un des principaux critères de choix de l'outil est lié à sa capacité à permettre la saisie, le stockage, l'extraction, et la publication des données. Cela inclut notamment la possibilité de saisir des informations spécifiques aux fiches (licences, par exemple) ou celle de publier ces informations dans différents formats (version web, imprimable, PDF, ou RSS).

De plus, l'outil doit avoir la possibilité de gérer des métadonnées (notamment des mots-clés) afin de pouvoir effectuer des recherches efficaces ou de filtrer l'information de façon thématique.

3.6 - Autres fonctionnalités

Des fonctionnalités plus "standards" sont attendues, citons notamment :

- gestion de logs d'activités et de statistiques des visites
- possibilité de lier l'outil à un SSO et/ou un annuaire LDAP
- interface d'administration complète, mais aisée à prendre en main

4 - Contraintes

Malgré l'ambition du projet et un cahier des charges relativement exigeant, les contraintes de la mise en exploitation des premières versions de PLUME sont assez fortes :

- techniques : l'outil doit reposer sur une solution LAMP (Linux, Apache, MySQL, PHP)

- humaines :
 - la prise en main de l'outil doit être aussi simple que possible afin de faciliter et limiter les temps de formation.
 - l'équipe de développement étant réduite (1 temps plein, avec la possibilité de faire appel à des experts internes), les développements spécifiques doivent être limités.
 - les compétences techniques internes sont existantes mais ne doivent pas faire l'objet d'une trop grande sollicitation.
- financières : le projet devra faire l'objet d'une excellente maîtrise des coûts, non seulement lors de sa mise en place, mais aussi à plus long terme pour une meilleure maintenabilité.
- temporelles : il s'agit probablement de la plus forte contrainte, puisqu'il a été décidé une mise en exploitation courant septembre 2007 sachant que la présente étude a débuté mi-mai 2007). Cependant, cette première mise en exploitation sera immédiatement suivie de la mise en place d'un processus d'amélioration visant à satisfaire les besoins non prioritaires qui n'auraient pas pu l'être sous peine de retarder le lancement du projet.

Ces ressources limitées ont fortement influencé le choix final de l'outil, en poussant l'équipe au pragmatisme (un cahier des charges allégé, volonté d'utiliser des outils préexistants plutôt que de faire du développement spécifique, choix de la simplicité et de l'efficacité plutôt que de l'exhaustivité en terme de fonctionnalités).

5 - Méthodologie

L'une des caractéristiques principales du projet PLUME est de s'inspirer du modèle de développement de projets de logiciels libres.

Ainsi, certains choix dans la mise en place du projet ne manqueront pas de rappeler les méthodes de développement préconisées dans l'essai d'Eric S. Raymond « *La cathédrale et le bazar* »³.

Par exemple la volonté de :

- fixer des échéances brèves et de fonctionner par itérations plutôt que de se donner une date de livraison pour un produit qui serait "fini"
- impliquer dès le départ des "utilisateurs" du projet, afin d'élargir les points de vues et d'avoir des retours permanents
- réutiliser des produits existants plutôt que de se lancer dans la conception d'un nouvel outil
- accepter que la première version du produit sera loin d'être parfaite (mais que chaque version en préfigurera une suivante plus efficace)
- se concentrer sur les contenus et les services plus que sur l'outil
- ne pas chercher la surenchère en terme de fonctionnalités, en visant plutôt la simplicité et une démarche pragmatique

Le choix du CMS s'est déroulé en 4 étapes principales :

1. **définition des objectifs** de la plate-forme (dans les grandes lignes)
2. **mise en place d'une maquette** (intégrée au sein du site web de l'UREC, fonctionnant sous SPIP). Cette maquette aura permis d'identifier les grandes caractéristiques fonctionnelles

³ http://www.linux-france.org/article/these/cathedrale-bazar/cathedrale-bazar_monoblock.html

attendues et, de par son caractère public, d'attirer des contributeurs intéressés à participer activement au projet. Sur cette maquette, ont été ajoutés au CMS SPIP un outil de workflow (Eventum) et un Wiki pour le travail collaboratif.

3. rédaction des spécifications fonctionnelles

4. **évaluation et sélection d'un outil** pour remplacer la maquette (SPIP, Eventum, Wiki), suivant les caractéristiques définies.

Nous allons ici détailler la quatrième étape, qui nous a conduit au choix d'un CMS pour PLUME.

D'abord, sachant qu'il existe plusieurs centaines de CMS et qu'il serait évidemment impossible de tous les tester, il a fallu faire une première présélection d'outils. Étant donné le temps disponible pour l'ensemble de la période d'évaluation, cette première présélection ne pouvait pas excéder une quinzaine d'outils. Celle-ci s'est basée sur des critères techniques (langage utilisés pour le développement de l'outil) et légaux (licence), puis sur une rapide évaluation des candidats potentiels (ceux qui disposaient d'un nombre conséquent d'installations en production, d'une large base d'utilisateurs, etc.) en essayant, si disponible, de tester une version de démonstration.

Cela nous a amené à faire une première sélection de 14 outils

Ensuite, ces applications ont été installées sur un serveur local, afin de pouvoir les tester en "conditions réelles". Cette évaluation ne s'est pas arrêtée aux fonctionnalités, puisqu'elle prenait en compte :

- la qualité de la documentation
- la qualité du support
- le type et la qualité du code utilisé
- une évaluation de la courbe d'apprentissage du logiciel
- la pérennité de l'outil
- et évidemment son adéquation à PLUME

Cela a permis de répartir les applications en différentes catégories, afin de mieux cerner leurs points forts et leurs points faibles dans le contexte du projet :

- CMS communautaires
- CMS institutionnels
- CMS éditoriaux
- Content Management Framework

Aucune application ne correspondait qu'à une et une seule catégorie, certaines pouvaient même être classées dans les quatre simultanément. Cependant cette classification a permis de dégager les principaux types de fonctionnalités attendues.

Les applications arrivées premières de chacune des catégories furent alors testées de façons plus complète, notamment en essayant de les utiliser comme une nouvelle maquette PLUME. Deux outils (Joomla et Drupal) étant arrivés ex aequo de leur catégorie, ce sont donc 5 CMS qui ont été évalués sur les critères suivants :

- Performances
- Flexibilité
- Extensibilité
- Sécurité
- Pérennité
- Intéropérabilité
- Simplicité d'utilisation
- Simplicité de configuration et d'administration
- Adéquation à la contrainte temporelle

Au final, seules 2 applications semblaient aptes pouvoir être rapidement mises en production tout en respectant le cahier des charges PLUME : Drupal et SPIP.

Ces deux applications ont donc été confrontées aux principales exigences des spécifications fonctionnelles (plus de trente critères).

Cette dernière évaluation, qui aura duré 3 jours, nous a finalement conduits à retenir Drupal. Plus modulaire et extensible que la plupart de ses concurrents, sa prise en main est relativement rapide et agréable. Sa large communauté d'utilisateurs et la présence de modules répondant à certaines problématiques de PLUME ont été déterminantes dans le choix de cet outil pour servir de plateforme au projet PLUME.

6 - De la difficulté d'une première sélection : le choix

6.1 - Critères de sélection

La première difficulté a été le fait de devoir effectuer une sélection parmi l'offre pléthorique de CMS existants. Ainsi, le site CMS Matrix⁴ en référence plus de 250 et il est couramment admis qu'il existe plus de 700 systèmes de management de contenu sur le marché.

Les premiers critères de choix ont donc été :

- technologie employée : l'outil doit nécessairement reposer sur PHP/MySQL
- licence : l'outil doit être sous licence libre, non seulement pour correspondre à l'objet du projet, mais surtout pour permettre les inévitables adaptations nécessaires.
- nombre d'installations en production : le nombre d'installations doit être suffisamment important afin d'éviter d'amener le projet dans une impasse technique. Comme il n'existe pas réellement de chiffres propres au nombre d'installations, il a été décidé d'évaluer le nombre et l'évolution du nombre de téléchargements⁵
- valeur ajoutée par rapport aux produits "proches" : beaucoup de produits étant très similaires en terme de fonctionnalités, et souvent issus de forks (reprise d'un code existant servant de base à un nouveau projet logiciel) d'autres projets opensource existants. Par exemple Xoops et e-xoops, phpnuke et postnuke, Mambo et Joomla, etc. Une rapide évaluation en se basant sur des recherches ou les spécifications publiques des logiciels a donc été faite afin de les

4 <http://www.cmsmatrix.org>

5 Exemple pour Joomla : <http://joomlancode.org/gf/project/joomla/reporting/?action=Stats>

départager.

- tests rapides : non éliminatoires, de rapides tests ont été effectués, soit sur les versions de démonstration des logiciels, soit (si disponible) sur les versions installées sur le site OpenSourceCMS⁶.

A l'issue de cette première sélection, 14 outils furent retenus :

- Drupal
- e107
- eZpublish
- Jaws
- Joomla
- Lodel
- MODx
- MySource
- SPIP
- Textpattern
- Tiki CMS
- Typo3
- Xaraya
- Xoops

6.2 - Note sur le cas de HAL

Nous avons envisagé d'utiliser une occurrence de HAL (<http://hal.archives-ouvertes.fr/>) comme plate-forme pour Plume : cet outil présente de nombreux points d'adéquation avec les fonctionnalités recherchées pour Plume et l'équipe HAL était prête à coopérer activement à une adaptation. Cependant, d'une part, la partie outil collaboratif entre membres de la communauté, nécessaire à Plume est très peu couverte par les fonctionnalités de HAL et cela aurait donc nécessité l'interfaçage de plusieurs outils, d'autre part, le format de publication des fiches doit être souple et HAL ne permet pas autant de possibilités qu'un CMS. Pour ces raisons, il nous a semblé que HAL n'est pas actuellement l'outil le plus adapté pour Plume.

Adéquation de HAL :

Bonne adéquation :

- recherche par mots-clés
- plusieurs catégories de mots-clés
- droits de MAJ et consultation fins
- modération des mises à jour
- interface utilisateur final complètement paramétrable dans la mesure où l'utilisateur final s'abonne autant de fois qu'il souhaite avec autant de profils différents ce qui lui permet de consulter HAL avec des interfaces différentes.
- MAJ automatique des dates de MAJ par fiche (vue récente sur la dernière, historisation des autres)
- classification automatique quand mise à jour
- nouveautés du site présentées de façon automatique sur une page nouveautés

⁶ <http://www.opensourcecms.com>

- analyse du contenu des meta données (champ obligatoire ou non, ...)

Nécessite une adaptation ou solution de contournement :

- gestion des logos pour les caractéristiques du logiciel
- rappel automatique aux contributeurs et responsables de rubriques après recherche par date de dernière mise à jour des fiches qui nécessitent d'être actualisées. À écrire.
- Base de données des membres du projet, des contributeurs, des responsables de rubriques : elle existe dans HAL mais n'est pas accessible à travers des vues déjà réalisées. Une solution serait d'avoir accès direct à la base, une autre de réaliser les vues utiles à Plume.

Pas prévu et/ou très compliqué à ce jour :

- HAL n'est pas prévu pour être un lieu d'échanges et permettre ainsi la création d'une communauté, la partie commentaires liés aux fiches, listes de discussions et forums spécialisés (à accès restreint ou non), wiki, etc. doivent être gérées à l'extérieur de HAL mais un lien vers l'url serait possible
- possibilité d'écrire au contributeur avec accusé de réception ou trace : on peut écrire au contributeur mais la solution proposée est un mailto qui va ouvrir le client de messagerie de l'utilisateur donc, pas d'AR possible
- vérification des urls et de l'orthographe impossible.
 - Url difficile à vérifier automatiquement : l'url peut être correcte (la page existe) mais fautive (ce n'est pas celle qu'on voulait).
 - Orthographe pas prévu

7 - Seconde sélection : tests

Ces outils ont été installés sur un serveur de tests local, afin de pouvoir les tester dans les meilleures conditions.

Étant donné le temps disponible pour l'évaluation (environ une dizaine de jours ouvrables), les applications ne furent pas testées de façon approfondie, mais plutôt comparées les unes aux autres afin d'en dégager les principales spécificités et les possibilités d'adéquation au projet PLUME.

Les tests peuvent s'avérer incomplets dans le cas où un critère bloquant a été rencontré.

Les principaux critères furent :

- présence d'une documentation structurée et complète (et éventuellement francophone)
- évaluation du support (forum ou mailing-liste) : dans des projets tels que PLUME, il est essentiel d'avoir un support réactif et efficace, afin de pouvoir prendre les bonnes orientations en cas de développements personnalisés ou, plus simplement, de pouvoir résoudre les problèmes et bugs rencontrés. L'importance et la réactivité du support ont donc été prises en compte (nombre de messages, activité quotidienne, délai de réponse, qualité de la réponse, etc.)
- une rapide évaluation du type et de la qualité du code (commenté ou non ? programmation objet ou non ? etc.)
- évaluation de la courbe d'apprentissage (présumée ou annoncée) : étant donné le délai très court de mise en place du projet, combien de temps faut-il pour :
 - appréhender les principales fonctionnalités ?
 - mettre en place les bons plugins ?
 - développer un plugin spécifique ?

- créer/adapter une charte graphique ?
- être efficace dans la mise en place d'une première version du projet ?
- une évaluation de l'accessibilité : la prochaine publication du RGAA (« Référentiel Général d'Accessibilité pour les Administrations »⁷) imposera un certain nombre de contraintes techniques afin de s'assurer du bon accès au site PLUME par des publics handicapés. Il faut donc anticiper cette problématique en la prenant en compte en amont du projet. Il s'agit ici d'une évaluation rapide, car la phase de tests et de mise en conformité peut être relativement longue et se fait en parallèle de la phase de réalisation du site.
- la pérennité de l'outil : difficile de prédire quels outils opensource seront toujours actifs dans un an ou deux, tellement le mode de développement collaboratif de ces applications tient à des facteurs difficilement identifiables. Cependant, certains éléments permettent de donner des indications sur une pérennité présumée :
 - date de la première version
 - date de la dernière version
 - nombre de téléchargements total
 - évolution (positive ou négative) du nombre de téléchargements dans le temps
 - taille de la communauté
 - qualité du support (délais et pertinences des réponses sur les forums ou les listes de diffusion)
 - historique du projet (motivation des auteurs initiaux)
 - cohésion et taille de l'équipe de développement actuelle
 - l'activité du dépôt CVS/Subversion en terme de modifications
 - les changelogs (patches de sécurité, corrections mineures ou majeures, qualité et pertinence des fonctionnalités ajoutées)
 - présence d'une roadmap publique cohérente et tenue à jour (même si elle n'est pas respectée)

Tous ces critères n'ont pu être évalués de façon complète pour chaque application, faute de temps, mais vous trouverez en annexe les principaux indicateurs et avis pour les 15 applications retenues.

8 - Catégorisation et classement

NB : cette classification se fait par rapport à l'analyse de besoins et des spécifications fonctionnelles PLUME, elle ne présume en rien des qualités intrinsèques de chaque application (dont la sélection est variable pour chaque projet).

Les applications ont été réparties en quatre catégories principales :

1. Les CMS "communautaires" : plutôt dédiés à la création de communauté, ils permettent de structurer le contenu avec une vision "**visiteurs/utilisateurs**".
2. Les CMS "institutionnels" : plutôt orientés "**portails**", ils sont extrêmement riches en fonctionnalités et permettent de gérer la quasi-totalité des cas de figures. Mais cela se paie par une courbe d'apprentissage longue (plusieurs semaines/mois annoncés⁸) et un temps de

⁷ <http://rgaa.referentiels.modernisation.gouv.fr/>

⁸ <http://typo3.org/about/new-to-typo3/>

mise en oeuvre non négligeable.

3. Les CMS "éditoriaux" : développés au départ sur les modèles des journaux en ligne, ils ont été conçus avec la volonté de privilégier/faciliter le travail des **rédacteurs d'articles** et le processus de publication.
4. Les CMF (Content Management Framework) : apparus plus récemment, il s'agit de "boîtes à outils" permettant de "développer son propre CMS", sur mesure. C'est donc avant tout le travail du **développeur** qui déterminera les fonctionnalités finales. Les perspectives offertes par les CMF sont intéressantes, du fait de leur extensibilité/modularité, mais la courbe d'apprentissage peut s'avérer longue. D'autant que chaque CMF a sa propre logique, et que la jeunesse relative de ces outils en fait des applications au développement parfois chaotique (alternance de frénésie de développement et d'absence de communication/releases, documentation incomplète, revirements et refonte des concepts, etc)

Cependant, il faut voir que chaque application peut toucher différentes catégories, et que ces dernières ne sont donc ni exclusives, ni "étanches". Ainsi, il serait faux de réduire un CMS "communautaire" à ses visiteurs ou un CMS "éditorial" à ses rédacteurs. Même s'il s'agit de choix de conception (ou historiques) qui influencent encore le développement et l'usage de l'outil aujourd'hui, aucune application ci-dessus n'est isolée dans l'une de ces catégories.

Le classement proposé est le suivant :

- CMS communautaires

1. Drupal
2. Joomla
3. e107
4. Xoops

- CMS insitutionnels

1. Typo3
2. eZpublish
3. tikiwiki
4. mySource

- CMS éditoriaux

1. SPIP
2. TextPattern
3. Lodel

- CM Framework

1. MODx
2. Xaraya
3. Jaws

9 - Troisième sélection : adéquation avec le projet PLUME

Les applications arrivées en tête de chaque catégories ont ensuite fait l'objet d'une évaluation plus complète basée sur l'adéquation potentielle avec le projet PLUME.

NB : 0 est la note la plus basse, 5 la meilleure.

Critère	Joomla	Drupal	SPIP	Typo3	MODx
Performances*	3	4	4	3	4
Flexibilité (interne**)	3	3	4	5	5
Extensibilité (externe***)	5	4	4	5	3
Sécurité	3	4	3	4	2
Pérennité****	4	5	5	4	3
Interopérabilité	3	4	4	3	5
Simplicité d'utilisation	4	5	5	3	4
Simplicité de configuration/admin	3	5	4	2	3
Adéquation à la contrainte temporelle	4	4	5	2	3
Total (sur 45)	32	38	38	31	32

*Inclus aussi la stabilité ou la capacité de l'application à ne pas s'effondrer dès qu'on ajoute une extension

**Comprendre les capacités "natives" de la plateforme

***Modules/plugins préexistants et capacité d'en créer de nouveau

****Efficacité du support, taille de la communauté, perspectives, etc. Mais aussi capacité à être repris/maintenu avec les compétences internes

Attention, il convient de rappeler une nouvelle fois que les fonctionnalités sont évaluées en fonction de leur adéquation avec le projet PLUME. Ainsi et par exemple, une évaluation de "2" pour la sécurité de MODx ne signifie pas que ce dernier est moins sécurisé que Typo3 (évalué à 4), mais qu'étant donné la relative jeunesse de MODx et le nombre d'occurrences installées en production plus faible, nous devons prendre en compte qu'une faille de sécurité soit découverte et traitée moins rapidement (par exemple via le système d'alerte du CERT).

Typo3 a été écarté pour sa courbe d'apprentissage bien trop longue par rapport aux contraintes temporelles de mise en place du projet.

MODx présente, sous la forme d'un framework, un concept innovant de CMS. Malheureusement, il est encore trop difficile d'évaluer la pérennité de cette application (dont une refonte est d'ores et déjà en cours).

Joomla est un des CMS le répandu sur le web et a remporté plusieurs récompenses. Mais les lourdeurs de son code, les doutes sur la pérennité de la compatibilité des extensions avec la nouvelle version de l'application, et son support certes très important quantitativement mais qualitativement moins performant que celui de certains de ses "concurrents" nous ont finalement amené à lui préférer Drupal. Paradoxalement, Joomla est ici victime de son succès : il est parfois difficile de s'y retrouver dans ses 1600 extensions et dans la profusion d'informations qui gravitent autour de cette application.

10 - Choix final

Au final, l'hésitation fut assez longue entre SPIP et Drupal, deux produits conceptuellement très différents dans le monde des CMS.

Le tableau suivant donne une idée (certes imprécise) des qualités des différents outils. On peut y distinguer notamment le point fort de SPIP dans sa capacité à publier et structurer rapidement l'information, face à un Drupal plus modulaire, plus souple, et probablement plus adaptable si l'on

est prêt à consacrer du temps à sa bonne configuration.

NB : 0 est la note la plus basse (fonctionnalité non implémentée), 5 la meilleure (fonctionnalité très satisfaisante). Un "P" signifie que l'utilisation d'un plugin est recommandée ou nécessaire pour cette fonctionnalité. Un "D" implique qu'un développement est nécessaire. Les critères sont les fonctionnalités demandées dans le cahier des charges (FT : Fonction transversale, FP Fonction principale, FC Fonction complémentaire)

Critères	Importance dans le projet PLUME	Drupal	SPIP
FT-1:Workflow			
FPrel-2:possibilité de corriger des contenus en rédaction	4	4	3
FPSg-1:respect des procédures de workflow définies	5	5D	2D
FPSg-10:actions programmées sur événements	3	4P	3D
FT-2:ACL			
FPMc-2:accès à des contenus en accès restreint	3	5	3P
FPRc-2:droit de publication limité	3	5	5
FT-3:Collaboration			
FPCo-5:possibilité de signaler une erreur dans un contenu	3	3D	3D
FPre-5:possibilité de compiler un contenu à partir d'autres	3	0D	0D
FCCo-1&FCSg-2:rédaction collaborative de contenu	3	4	4
FCCo-3:forums restreints	2	5P	4
FCCo-6:outils de travail collaboratif	4	4P	4P
FT-4:Manipulation de données			
FPCI-1:Fiches imprimables	5	4P	5
FPCI-2:recherche par mots clés	4	?	4D
FPCI-2:recherche Full-text	4	?	5
FPCI-3:abonnements fils RSS	4	4	4
FPCI-4:publication de contenus/docs	5	4	5
FPCI-1:Fiches classées par rubriques	5	4	5
FPCo-1:rédaction de contenu selon modèle de données (ex:fiches)	5	5P	3D
FPSg-3:gestion de mots-clés / metadonnées	5	5	5
FPSg-4:multi-rubriquage	4	5	4D
FT-5:Autres fonctionnalités			
FPre-11:statistiques détaillées	3	4	5
FPCe-2:Interface d'administration	3	5	5
FPSg-2:enregistrement de membres	3	5	5
FPSg-2:liaison avec annuaire LDAP	3	5P	5P
FPSg-2:authentification par SSL/certificats	3	?	?
FPSg-2:SSO	3	?P	?P
FPSg-5:gestion de statistiques et indicateurs	3	3D	4
FPSg-7:identification des URL inaccessibles	2	0D	0D
FPSg-8:historique des actions sur les contenus	3	4P	4
FPSg-9:gestion de flux RSS externes	3	5	5
FPSg-9:gestion de listes de diffusion	2	4P	4P
FCCI-2:forum(s) liés aux contenus	4	4	4
FCSg-1:localisations en plusieurs langues	2	?	5

D'un côté, SPIP bénéficie d'un excellent support francophone, d'un développement actif et d'une bonne quantité d'installations (et donc d'utilisateurs et d'experts) au sein de la communauté. D'un autre côté, malgré un système de plugin performant et pratique, il lui manque encore quelques

fonctionnalités indispensables au projet PLUME (notamment la possibilité de gérer proprement des champs externes, car les "champs extra" ne sont qu'une astuce technique, mais peu pérenne et techniquement limitée).

Au contraire, Drupal est une application bien moins répandue dans la communauté. Plus difficile d'accès, sa courbe d'apprentissage est plus longue et l'outil peut se montrer par certains côtés déstabilisant pour les personnes habituées aux CMS classiques (notamment sa gestion de la taxonomie).

Cependant, il faut reconnaître qu'une fois installés quelques modules additionnels, Drupal se révèle un outil extrêmement complet et efficace. Ainsi, sans développement spécifique, il répond à près de 80% du cahier des spécifications fonctionnelles PLUME, et sa structure modulaire et son API bien documentée permettent d'espérer une mise en place relativement rapide des fonctionnalités manquantes.

11 - Conclusion

Avant de conclure, il convient une nouvelle fois de rappeler que les critères qui auront mené au choix de Drupal correspondent à la volonté de mettre en place un outil en adéquation avec le cahier des charges du projet PLUME, en tenant compte des contraintes techniques et temporelles assez fortes. Cette sélection ne présume donc en rien de la qualité d'autres CMS dans des contextes différents.

L'un des enseignements de cette rapide évaluation de CMS démontre que la principale difficulté n'est pas tant de choisir un outil parmi d'autres, mais d'essayer d'obtenir une bonne lisibilité dans l'offre pléthorique des CMS. En effet, les CMS sont des outils relativement rapides à développer, et la tentation est forte de vouloir en développer un "sur mesure" pour son propre besoin. Cependant, ce choix est aujourd'hui contre-productif (on réinvente la roue, on se heurte à des problèmes déjà résolus par d'autres).

Un autre enseignement est que cette offre est suffisamment vaste pour qu'une très grande part des besoins soient couverts nativement, pour peu que l'on accepte de consacrer plusieurs jours à tester différents outils. Ainsi, le concept de Content Management Framework permet de réaliser des CMS aux fonctionnalités spécifiques en s'appuyant sur des briques fonctionnelles préexistantes.

De plus, les meilleurs CMS offrent des mécanismes d'extensibilité performants, le plus souvent sous forme de plugins additionnels développés par des tiers ou, plus rarement, par la possibilité d'intégrer des portions de code PHP/SQL directement au sein de l'application (sans modifier les fichiers sources et s'orienter de ce fait vers un fork difficilement maintenable).

Au final, le choix de Drupal nous semble être le meilleur pour le projet PLUME car il s'agit d'un outil stable, actif, bien documenté, pérenne, extensible, et suffisamment souple pour permettre les inévitables adaptations spécifiques au projet PLUME.

Annexes

1 - Glossaire et définitions

NB : la plupart de ces termes sont des anglicismes, mais leurs équivalents français sont très peu utilisés.

ACL : Access Control Lists, ou ACL : listes de contrôle d'accès.

Dans le cas d'un CMS, les ACL permettent de gérer les droits d'accès des utilisateurs d'un site à certaines fonctionnalités de celui-ci.

CMS : Un Content Management System (SGC: Système de gestion de contenus) est une application permettant de créer, produire et gérer rapidement des contenus (textes, images, etc), le plus souvent pour des sites Internet. Ils permettent d'en réduire le temps de programmation en intégrant des fonctionnalités génériques (gestion des auteurs et utilisateurs, publication du contenu, etc.)

Fork : Littéralement « fourche », fork est une commande UNIX permettant d'engendrer un processus « enfant ». Par extension, un fork d'application est la création d'une nouvelle version de cette application en se basant sur le code source de la première, afin de lui adjoindre des fonctionnalités.

Framework : Un cadre d'applications informatique (en anglais, Application Framework) ou cadriciel est un ensemble de bibliothèques permettant le développement rapide d'applications. Ils fournissent suffisamment de briques logicielles pour pouvoir produire une application aboutie rapidement. Ces composants sont organisés pour être utilisés en interaction les uns avec les autres. Ils sont en principe spécialisés pour un type d'application.

(Source : <http://fr.wikipedia.org/wiki/Framework/>)

Plugin (addon, module, greffon, extension) : développement plus ou moins important, qui vient se « greffer » sur l'application de base afin d'étendre ses fonctionnalités.

Template : Un template est un moyen de séparer le fond (le contenu informationnel) de la forme (la manière dont il est présenté). On parle aussi de « patron » ou « gabarit » ou « modèle » quand il ne s'agit que de la couche graphique, mais la notion de template recouvre aussi le moteur technique (« template engine ») chargé d'intégrer le contenu dans la couche graphique ou de gérer le cache.

Très utilisé dans la conception de sites web, un template agit comme un modèle dans lequel seuls certains éléments sont modifiables (le contenu). Cela facilite la conception et la mise à jour des sites, aussi bien sur le contenu que sur la présentation.

- changer la charte graphique du site revient à changer le template et cela met à jour toutes les pages du site ;
- ajouter une page ne consiste plus qu'à en écrire le contenu.

(Source partielle : <http://fr.wikipedia.org/wiki/Template>)

Workflow : Un workflow est un système permettant d'automatiser un flux d'informations au sein d'une organisation, par exemple en transmettant automatiquement des documents entre des personnes. (Source <http://fr.wikipedia.org/wiki/Workflow>)

2 - Résultats de la seconde évaluation

2.1 - Drupal

- Site officiel : <http://drupal.org/>
- Démo opensourceCMS : http://opensourcecms.com/index.php?option=com_content&task=view&id=132
- Grille CmsMatrix : <http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1050>
- Version évaluée : 5.1
- Documentation en ligne : <http://api.drupal.org/api/HEAD>
- Support en ligne : <http://drupal.org/forum>
- Licence : GPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : très importante en anglais
 - date de première / dernière release : 15/01/2001 (v1.0) - 19/01/2007 (V5.1)
 - communauté francophone : oui, moyennement active : <http://drupalfr.org/>
 - taille de la source : 2.14Mo
 - changelog : [\[http://cvs.drupal.org/viewcvs/drupal/drupal/CHANGELOG.txt?view=markup \]](http://cvs.drupal.org/viewcvs/drupal/drupal/CHANGELOG.txt?view=markup)
 - activité du développement : très forte <http://cvs.drupal.org/viewcvs/drupal/drupal/>

Informations complémentaires et avis dans le cadre du projet

Drupal est à mi-chemin entre le CMS et le CMF (Content Management Framework).

Sa courbe d'apprentissage est réputée moyenne (quelques jours pour les bases, quelques semaines pour le système d'API).

L'un des principaux points forts de Drupal réside dans sa grande modularité. Paradoxalement, c'est sans doute ce point qui pénalise Drupal en terme de nombre d'installations en production. En effet, malgré une installation très simple, l'administrateur de la plateforme se retrouve avec une "coquille vide" dont il faut activer les modules suivant ses besoins (blog, flux RSS, etc). Ce temps de configuration est indispensable et la prise en main n'excède pas quelques heures, mais elle peut rebuter les personnes qui souhaite juste découvrir l'outil. De plus certains modules très utiles ne sont pas fournis dans le package de base, et doivent donc être téléchargés et installés séparément avant de pouvoir vraiment tirer bénéfice des fonctionnalités de Drupal.

Ressources intéressantes dans le cadre d'une utilisation de Drupal pour PLUME :

- Module Views (permet de définir des requêtes sur les contenus et de les afficher de façon structurée à l'utilisateur) :
- Documentation : <http://drupal.org/node/109604>
- Snippets : <http://drupal.org/node/47412>
- Module CCK (permet de définir des champs de saisie personnalisés) :
- Documentation : <http://drupal.org/node/101723>
- Liste d'extensions : <http://drupal.org/project/Modules/category/88>

- Module Drutex (LaTeX) : <http://drupal.org/project/drutex>
- Module taxonomy filter (permet la sélection de différents mots clés): http://drupal.org/project/taxonomy_filter
- Module diff (permet une visualisation des changements entre 2 versions d'un article) : <http://drupal.org/project/diff>

Points forts

- Disposant d'une large communauté (mais plus restreinte que celle de Joomla par exemple, du fait de sa plus grande complexité), Drupal est utilisé par de nombreux sites institutionnels depuis plusieurs années.
- Il dispose de nombreuses extensions, dont certaines sont identifiées comme pouvant aider à répondre aux besoins définis dans les spécifications fonctionnelles PLUME (notamment CCK, Views, Taxonomy, Workflow et Actions).
- Le code est relativement léger et bien documenté.
- Possibilité d'intégrer de façon très souple des "snippets" (portions de codes PHP/HTML/SQL) pour modifier dynamiquement le comportement des pages sans avoir à modifier les fichiers sources ou créer de plugins.
- L'organisation de contenus Drupal est basée sur l'utilisation du module Taxonomy : des "Catégories" ("Vocabularies") contenant une hiérarchie simple ou complexe de "Termes" ("terms", prédéfinis ou "tags" libres). C'est le même principe que les "mots-clés" classés par "groupe de mots clés" de SPIP
- A noter qu'à part pour les administrateurs, il n'existe pas vraiment de backend pour Drupal : lorsque l'on a accès en édition à un article, un bouton "éditer" apparaît. Il n'est donc pas utile de passer par l'interface d'administration pour l'utilisateur final, le seul fait d'être identifié lui permet de modifier directement son contenu.
- Le système de gestion de permissions est fin et complet (mais en conséquence assez long à configurer)

Points faibles

- Le système de template est plutôt complexe et pas forcément pratique (par rapport au système de boucles de SPIP, par exemple).
- Malgré l'installation du fichier de langue Fr, la traduction de Drupal est loin d'être complète. Or, son administration reste relativement complexe du fait d'un très grand nombre d'options disponible, et le fait de ne pas avoir de traduction peut rendre certains concepts encore plus difficile à appréhender (rien de rédhibitoire, mais tout dépend du public cible). Le phénomène est encore plus frappant lors de l'utilisation de plugins (peu d'entre eux étant correctement traduits).
- Pour certaines actions, l'utilisateur de la base MySQL doit disposer de privilèges avancés (comme CREATE TABLE), mais cela ne devrait pas être un problème dans le contexte technique de PLUME.
- Au niveau de la saisie des articles, messages de forum et commentaires, les options de mise en forme sont en fait basées sur du HTML (filtré ou non), ce qui ne permet pas une bonne séparation du fond et de la forme (il est possible d'ajouter l'éditeur TinyMCE [<http://tinymce.moxiecode.com/example.php>] mais cela ne règle pas le problème.

Cependant, le problème se pose pour les autres outils et n'est pas réellement rédhibitoire pour PLUME.

Drupal semble donc le meilleur candidat pour le projet PLUME.

2.2 - e107

- Site officiel : <http://www.e107.org/>
- Démo opensourceCMS :
[http://opensourcecms.com/index.php?option=com_content&task=view&id=133]
- Grille CmsMatrix : [<http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1158>]
- Version évaluée : 0.7.8
- Documentation en ligne : http://wiki.e107.org/?title=Main_Page
- Support en ligne : http://e107.org/e107_plugins/forum/forum.php
- Licence : GPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : importante en anglais
 - date de première / dernière release : 21/09/2004 - 17/02/2007 (v0.7.8)
 - communauté francophone : oui, mais peu/pas active : <http://etalkers.tuxfamily.org/>
 - taille de la source : 8.18Mo
 - changelog
 - activité du développement : forte : http://e107.org/show_cl.php - http://e107.org/show_cl.php?name

Informations complémentaires et avis dans le cadre du projet

e107 est à mon avis le CMS communautaire le plus rapide à mettre en place. Il comprend de nombreuses fonctionnalités intégrées, et son administration est claire (à défaut d'être ergonomique : les options sont listées dans l'ordre alphabétique et non par groupe de fonctions). Mais le produit semble encore assez jeune, l'activité de correction de bugs importante, et l'organisation de la communauté un peu brouillonne (par exemple, je n'ai pas réussi à trouver l'un des plugins indiqué dans la documentation sur le site).

e107 dispose donc d'un bon potentiel, notamment si l'on souhaite créer un site internet très rapidement, mais sa valeur ajoutée par rapport à des concurrents comme Drupal n'est pas suffisamment importante pour compenser le risque d'une perte de vitalité due à sa relative jeunesse.

Points forts

- Une communauté anglophone active
- Installation rapide et simple
- De nombreuses extensions
- Possibilité d'ajouter ses propres champs (voir http://wiki.e107.org/?title=Content_Management_Personal_Content_Manager)

Points faibles

- Pas de réelle valeur ajoutée dans le cadre de PLUME
- Un effet "mode" ou "tendance" (sans réelle argumentation derrière) qui pourrait faire douter de sa pérennité.
- Un outil relativement jeune
- Système de template peu pratique comparé à d'autres

2.3 - eZpublish

- Site officiel : <http://ez.no/>
- Démo opensourceCMS :
[http://opensourcecms.com/index.php?option=com_content&task=view&id=182]
- Pas de démo locale, à cause de la version de PHP requise
- Grille CmsMatrix : [<http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1063>]
- Version évaluée : 3.9.2
- Documentation en ligne : http://ez.no/doc/ez_publish
- Support en ligne : <http://ez.no/community/forum>
- Licence : GPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : importante en anglais
 - date de première / dernière release : ? - 4/05/2007 (v3.9.2)
 - communauté francophone : oui, mais peu active : <http://ezpublish-france.fr/>
 - taille de la source : 55Mo (!)
 - changelog : http://ez.no/download/ez_publish/changelogs/
 - activité du développement : forte, mais il semblerait que le développement du noyau soit avant tout réalisé par la société eZsystems et non par la communauté.

Informations complémentaires et avis dans le cadre du projet

La documentation accessible est de mauvaise qualité. Il semblerait que pour obtenir des informations complémentaires, il faille passer par les services de formation d'EzSystems.

L'installation peut être particulièrement difficile si on ne veut pas utiliser le « bundle Windows » (ensemble logiciel contenant l'application, mais aussi le serveur web, la base de données, etc.) . Ainsi, EzPublish ne fonctionne pas avec PHP 5 (sorti pourtant à l'été 2004 !). Une version compatible PHP6 est annoncée, mais aucune date ferme ne devrait être prise en compte, étant donné le caractère libre des développements. L'application réclame une disponibilité de 64Mo de RAM au niveau du php.ini pour tourner (là où la plupart des autres se satisfont de 8 ou 32Mo).

Le bundle Windows ne semblait pas fonctionner dans l'arborescence choisie. Après plusieurs tentatives, j'ai dû l'installer dans l'arborescence par défaut. Ensuite, ce bundle installe Apache et MySQL sous forme de services lancés automatiquement au démarrage. Cela n'est pas très sécurisé ni très pratique si on veut pouvoir les lancer ou les arrêter rapidement et simplement, par exemple dans le cas où l'on utiliserait d'autres installations Windows/Apache/Mysql/PHP. Le bundle Typo3 est beaucoup plus souple.

L'utilisation de l'interface d'administration m'a paru particulièrement lente (plusieurs secondes entre un clic et l'affichage de la page demandée).

Par contre, cette interface d'administration est riche et beaucoup mieux pensée que celle de Typo3 (plus ergonomique, plus claire, plus lisible).

La logique d'eZpublish fait que chaque contenu est un "objet". Cela en fait un outil extrêmement personnalisable.

Malheureusement, cette richesse fonctionnelle se paie par une courbe d'apprentissage du Framework qui me parait disproportionnée par rapport aux objectifs de PLUME.

eZpublish dispose d'extensions dans différents domaines : [<http://ez.no/community/contribs>]. Cependant celles-ci ne sont pas très nombreuses par rapport aux autres produits étudiés. L'installation de l'extension "wiki" (la seule identifiée comme intéressante pour le projet PLUME, s'est soldée par un échec du fait que l'application tournait sous Windows).

Après plusieurs heures de tests relativement intensifs, je retire de l'expérience avec eZpublish le sentiment que l'application possède certes un très fort potentiel, mais que celui dépasse largement le cadre du projet PLUME et surtout que sa mise en oeuvre (complexe + contraintes techniques + courbe d'apprentissage) ne parait pas adaptée au projet.

Points forts

- Richesse des fonctionnalités
- Souplesse du framework
- Manipulation de documents bureautique (import/export MS Word, OpenOffice.org, etc..)
- Interface d'administration bien pensée, malgré le grand nombre d'options

Points faibles

- Incompatibilité PHP5 !
- Courbe d'apprentissage de plusieurs semaines/mois
- Installation et fonctionnement sous Windows peu facilitée
- Documentation accessible insuffisante

2.4 - Jaws

- Site officiel : <http://www.jaws-project.com/>
- Démo opensourceCMS : [http://opensourcecms.com/index.php?option=com_content&task=view&id=2157]
- Grille CmsMatrix : [http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=zXnAhiPQAUMci6F4Uk_cdw]
- Version évaluée : 0.7.1
- Documentation en ligne : <http://wiki.jaws-project.com/doku.php>
- Support en ligne : <http://www.jaws-project.com/?page/contribute> (mailing lists, pas de forum)
- Licence : LGPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : assez faible (effet mailing-list versus forums ?), en anglais
 - date de première / dernière release : 29/03/2005 (v0.5) - 19/03/2007 (v0.7.1)
 - communauté francophone : non
 - taille de la source : 10-15Mo (suivant les modules installés)
 - changelog : <http://dev.jaws-project.com/cgi-bin/trac.cgi/timeline>

- activité du développement : moyenne : <http://dev.jaws-project.com/cgi-bin/trac.cgi/browser>

Informations complémentaires et avis dans le cadre du projet

Globalement, je trouve Jaws plutôt adapté à des plates-formes de publications "simples", notamment plates-formes de blogs. L'application est propre (code, graphisme, fonctionnalités) mais encore un peu "fraîche" et buguée (erreurs javascript/ajax dans l'interface d'admin).

Jaws n'est, en tant que tel, pas un mauvais candidat, mais l'application nécessiterait de développer de nombreuses extensions pour être en adéquation avec le cahier des charges PLUME.

Points forts

- repose sur une architecture MVC
- application récente, au code propre
- design et ergonomie agréable
- bonne modularité

Points faibles

- templating limité
- accessibilité moyenne (partiellement lié au point précédent)
- encore pas mal de bugs (notamment javascript)
- communauté assez restreinte, pas forcément très accessible
- pérennité fragile

2.5 - Joomla

- Site officiel : <http://www.joomla.org/>
- Démo opensourceCMS : [\[http://opensourcecms.com/index.php?option=com_content&task=view&id=2097\]](http://opensourcecms.com/index.php?option=com_content&task=view&id=2097)
- Grille CmsMatrix : [\[http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=VRQptHQyGsFCFf8IsmDB3g\]](http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=VRQptHQyGsFCFf8IsmDB3g)
- Version évaluée : 1.0.12 (principalement) et 1.5.0 (pour tester la compatibilité des modules)
- Documentation en ligne : <http://help.joomla.org/>
- Support en ligne : <http://forum.joomla.org/>
- Licence : GPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : très importante
 - date de première / dernière release : 17/09/2005 (v1 suite au fork avec Mambo) - 25/12/2006 (v1.0.12)
 - communauté francophone : oui, importante : <http://www.joomlafrance.org/>
 - taille de la source : 7Mo
 - changelog : <http://www.joomla.org/content/category/5/34/78/>

- activité du développement : très forte

Informations complémentaires et avis dans le cadre du projet

- Tutoriels : [<http://www.joomlafacile.com/content/view/83/32/>] (notamment un sur les ACL)
- Liste des extensions : <http://extensions.joomla.org/>

Joomla est la principale référence open-source en terme de CMS communautaire. Issu d'un fork de Mambo (avec lequel la branche 1.0 reste compatible), sa communauté est la plus importante (plus de 100 000 membres), elle dispose de près de 1700 extensions, de plusieurs centaines de thèmes. Clairement orientée vers les utilisateurs n'ayant pas particulièrement de connaissances en PHP, son large support international, son interface agréable à l'oeil (sans être pour autant toujours pratique et bien pensée) et sa grande modularité en font un outil pertinent pour qui souhaite mettre rapidement en ligne un site web.

La version 1.5 est actuellement en phase beta 2, la release stable étant prévue pour la fin de l'année. Par contre, cette version introduit une large réécriture du code existant, ce qui pourrait signifier l'incompatibilité d'un très grand nombre de modules existants.

Soutenu et structuré par une organisation sans but lucratif⁹ qui apporte des éléments de pérennité à l'application, Joomla est donc un candidat potentiel intéressant dans le cadre du projet PLUME.

Cependant, la qualité et la quantité de son code (plus de 65 000 lignes en grande partie héritée de Mambo, contre 9 000 pour Drupal), en fait un outil parfois difficile à maintenir avec une impression finale de "patchwork" d'extensions pas forcément cohérentes ou compatibles entre elles.

Dans le cadre de PLUME, cela signifie qu'il faudra probablement développer un certain nombre de composants pour Joomla, et que ceux-ci pourraient bien ne pas être compatibles avec la future version.

Pour ces raisons, le choix de Joomla me paraît finalement assez peu adapté pour le projet PLUME. Cela n'enlève rien aux qualités de cet outil facile d'accès et extensible, mais dont la valeur ajoutée dans le contexte PLUME n'égale pas celle d'autres outils.

Points forts

- Application "validée" par une base d'installations très conséquente
- Communauté extrêmement importante (au moins quantitativement)
- Installation guidée et simplifiée
- Interface d'administration simple à prendre en main
- Très grande extensibilité
- Nombreuses ressources disponibles (tutoriels, documentations, thèmes, etc)

Points faibles

- Trop d'extensions, de qualité très variables
- Ressources souvent dispersées

⁹ <http://www.opensourcematters.org/>

- Un code hérité d'une autre application, avec certaines lourdeurs

2.6 - Lodel

- Site officiel : <http://www.lodel.org>
- Démo officielle : [<http://lodeldemo.org/>]
- Grille CmsMatrix : [<http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=rpjuFWZfmXJm3az71xIXkA>]
- Version évaluée : 0.7
- Documentation en ligne : <http://www.lodel.org/sommaire30.html> (relativement complète et claire)
- Support en ligne : http://sourcesup.cru.fr/mail/?group_id=193
- Licence : GPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : restreinte
 - date de première / dernière release : 2001 (v0.1 ?) - 07/11/2006 (v0.7.5)
 - communauté francophone : oui (essentiellement francophone, d'ailleurs)
 - taille de la source : 3Mo
 - changelog : non, mais CVS sur sourceSup <http://sourcesup.cru.fr/projects/lodel/>
 - activité du développement : lente, mais régulière

Informations complémentaires et avis dans le cadre du projet

Informations

- Fonctionne avec un serveur Servoo : <http://www.servoo.net> - <http://sourceforge.net/projects/servoo>
- La gestion des utilisateurs comprend 4 statuts : visiteur, rédacteur, éditeur, administrateur (pas d'équivalent relecteurs, donc)
- Une version 0.8 est depuis longtemps disponible sur SVN, mais n'a jamais été publiée

Globalement, l'application semble très intéressante pour la publication de documents bureautiques (même si elle contraindrait à priori à utiliser Word ou des documents OOo 1.1), notamment via une bonne gestion des métadonnées et du modèle éditorial. Par contre, son développement ne semble pas très actif, et sa mise en oeuvre pas forcément adaptée au contexte PLUME (pas d'interaction avec les visiteurs ou membres de la communauté, nécessite la mise en place d'un serveur Servoo dont la stabilité serait à évaluer, pas d'ACL).

La communauté (quasi exclusivement francophone) est peu active (en terme de nombre de messages - 10 à 20 par mois), mais présente la caractéristique intéressante d'être du milieu enseignement et recherche.

Cela fait de Lodel un outil intéressant, mais aux fonctionnalités spécifiques. Il lui manque toutefois plusieurs fonctionnalités cruciales par rapport au projet PLUME et, malheureusement, Lodel semble assez difficile à étendre sans forker.

Points forts

- Le modèle éditorial semble relativement souple et puissant
- Utilise un moteur de template puissant (LodelScript, proche des squelettes SPIP)
- Facilité pour le rédacteur : Lodel est destiné à gérer des documents bureautique

Points faibles

- Installation sous Windows difficile. Installation sur LAMP beaucoup plus simple (il faut bien penser à appliquer les droits nécessaires sur certains dossiers)
- Lodel ne semble capable de manipuler que des fichiers bureautiques (hormis pour le métadonnées ou les "mini-textes"). Pas de formulaire en ligne.
- Servoo semble en redéveloppement, ou son développement semble arrêté (cf site servoo)
- Servoo ne semble pas pouvoir ouvrir un fichier .odt (alors que les .doc ou .sxw semblent passer correctement)

Lodel fait parfaitement ce pour quoi il a été conçu (gestion de revues en ligne), mais l'adéquation avec le projet PLUME semble difficile.

2.7 - MODx

- Site officiel : <http://modxcms.com/>
- Démo opensourceCMS :
[http://opensourcecms.com/index.php?option=com_content&task=view&id=2101]
- Grille CmsMatrix : [http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=g1Mtkd4OKHcmx4_PXiURYA]
- Version évaluée : 0.9.5
- Documentation en ligne : <http://modxcms.com/documentation.html>
- Support en ligne : <http://modxcms.com/forums/index.php>
- Licence : GPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : importante
 - date de première / dernière release : 28/04/2005 (v0.9, fork du CMS etomite.org) - 06/12/2006 (v0.9.5)
 - communauté francophone : oui, forum francophones réactifs sur <http://modxcms.com/forums/>
 - taille de la source : 7Mo
 - changelog
 - activité du développement : régulière, mais moyennement active

Informations complémentaires et avis dans le cadre du projet

Informations

- Comparaison entre ModX et Joomla/Drupal/SPIP :
[<http://modxcms.com/forums/index.php/topic,6873.0.html>] (lire notamment la comparaison

avec SPIP)

MODx est un Content Management Framework relativement récent. Bien plus complet que Jaws au premier abord, son API à l'air aussi bien plus puissante.

MODx comprend nativement un système de [http://wiki.modxcms.com/index.php/Adding_TV_Widgets_without_hacking_the_core Template Variables] qui permet d'ajouter des champs à l'interface de saisie. Malheureusement, si ces champs de saisie peuvent être facilement positionnés à l'affichage, cela ne semble pas être le cas dans l'interface d'administration/saisie.

La personnalisation du CMS final semble pouvoir être relativement fine une fois acquis les bases des "snippets" et des "chunks" qui composeront les différents éléments de la page. La snippet Ditto¹⁰ semble d'ailleurs indispensable pour pouvoir manipuler/imbriquer correctement les données.

Cependant, si MODx semble relativement complet côté outil, il semblerait bien qu'il y ait un point rédhibitoire dans le cadre de son utilisation dans le projet PLUME : l'absence de workflow¹¹ !

Il me paraît un peu risqué d'adopter pour PLUME une application dont le fonctionnement pourrait radicalement changer dans les mois qui viennent. Cependant, il faut reconnaître que MODx est bien pensé et conçu, et peut répondre à une bonne partie du cahier des charges PLUME si l'on est prêt à se donner les quelques semaines nécessaires à gravir la courbe d'apprentissage.

Points forts

- MODx est un vrai Content Management Framework, et non juste un CMS avec une API bien documentée
- Templating puissant (basé sur des micro-templates)
- Interface claire et agréable

Points faibles

- Version 1 en cours de développement, réécriture totale de la branche 0.9.x
- Du fait de la réécriture du code, la documentation actuelle n'est pas extrêmement fournie
- Courbe d'apprentissage sans doute trop longue par rapport aux contraintes temporelles du projet PLUME

2.8 - MySource

- Site officiel : <http://matrix.squiz.net/>
- Grille CmsMatrix : [<http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=27AkiDhQVfmIIJ9B8oCjZg>]
- Version évaluée : 3.12.4
- Documentation en ligne : <http://matrix.squiz.net/download/documentation>
- Support en ligne : <http://matrix.squiz.net/developer/support-forums>
- Licence : pour le noyau : choix entre GPL et Squiz Supported Licence¹² ; licences diverses

¹⁰ <http://ditto.modxcms.com/>

¹¹ http://modxcms.com/forums/index.php/topic_5835.msg41428.html#msg41428

¹² <http://matrix.squiz.net/evaluations/licence>

- (et notamment commerciales) pour les plugins
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : moyenne, voir faible
 - date de première / dernière release : 2000 (v1) ; 7/06/07 (v3.14.1)
 - communauté francophone : non
 - taille de la source : 14Mo
 - changelog : <http://matrix.squiz.net/developer/changelogs>
 - activité du développement : régulière, mais essentiellement dû à la société Squiz

Informations complémentaires et avis dans le cadre du projet

Il semblerait que le côté "commercial" de l'application soit assez fort. Ainsi, le moteur de recherche n'est pas fourni avec l'application mais vendu à part :

- <http://matrix.squiz.net/modules/all/search>
- <http://matrix.squiz.net/modules/choosing-gpl-or-ssv>

De plus, je n'ai pas trouvé trace d'une réelle communauté francophone, et le forum de support¹³ est assez peu actif et ce depuis plusieurs années¹⁴, ce qui est loin d'être bon signe.

Ces raisons m'ont paru rédhibitoires pour le projet PLUME, et l'application a donc été écartée du choix final.

Du coup, MySource est écarté de l'évaluation afin de pouvoir consacrer plus de temps aux autres projets.

2.9 - SPIP

- Site officiel : <http://www.spip.net/>
- Démo opensourceCMS : [\[http://opensourcecms.com/index.php?option=com_content&task=view&id=2081\]](http://opensourcecms.com/index.php?option=com_content&task=view&id=2081)
- Grille CmsMatrix : [<http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=PuPeVMWojB9mMI4klnRnw>]
- Version évaluée : 1.9.2
- Documentation en ligne : <http://www.spip.net/fr>
- Support en ligne : http://www.spip.net/fr_rubrique116.html
- Licence : GPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : importante (essentiellement francophone)
 - date de première / dernière release : 07/2001 (v1 : [\[http://www.spip.net/fr_article918.html\]](http://www.spip.net/fr_article918.html)) - 22/02/2007 (v1.9.2a)
 - communauté francophone : oui, très importante (cf news://gmane.comp.web.spip.user)
 - taille de la source : 9,6Mo
 - changelog : http://www.spip.net/fr_rubrique155.html
 - activité du développement : importante

¹³ <http://matrix.squiz.net/developer/support-forums>

¹⁴ http://matrix.squiz.net/developer/support-forums/integration/posts_submitted_by_month.png

Informations complémentaires et avis dans le cadre du projet

Les limitations de SPIP sont essentiellement côté Workflow et ACL, les 2 étant fixées par le noyau.

Cependant, le système de plugins introduit dans la version 1.9 semble pouvoir permettre de contourner ces limitations :

- Liste de plugins : <http://www.spip-contrib.net/+Plugin-+>
- Gérer de accès restreints par groupes d'utilisateurs
 - <http://www.spip-contrib.net/Le-plugin-acces-restreint-par> et <http://www.spip-contrib.net/Le-plugin-Acces-Restreint,1420>
 - http://p7app.geneve.ch:8007/spip/article.php3?id_article=288
- Gestion de champs personnalisés ?
 - <http://www.spip-contrib.net/Forms-tables-carnet-de-notes>
 - <http://article.gmane.org/gmane.comp.web.spip.user/93801/>
 - <http://www.spip-contrib.net/Personnaliser-les-champs-de-l>
- Travail collaboratif / wiki
 - Gribouille : <http://www.spip-contrib.net/Chantier-Gribouille-un-wiki>
- Recherche multicritères : <http://www.netdeveloppeur.com/tutoriaux-et-astuces-spip/spip-recherche-multicriteres-article34.html> et <http://www.spip-contrib.net/Un-tableau-d-articles-a-entrees>

SPIP semble capable de gérer des tables externes quasi-nativement. Cela pourrait résoudre le principal inconvénient, à savoir la non-adéquation des champs côté rédacteur avec les champs souhaités dans le cadre du projet PLUME.

Côté Workflow, les possibilités de SPIP sont assez restreintes, mais peuvent là aussi être étendue par plugins.

Enfin, SPIP propose désormais un certain nombre de fonctionnalités qui peuvent être intéressantes dans le cadre de PLUME, comme un wiki intégré, la possibilité de ne pas avoir à utiliser l'espace privé pour modifier des informations (tout en conservant la vérification des droits d'accès), la gestion des révisions (compatible avec l'utilisation de tables externes ?), le multi-rubriquage, l'export de données XML, gestion LDAP/OpenID, etc.

Sa communauté francophone est de loin la plus performante (qualitativement et quantitativement), cela en fait une solution pérenne et un choix à risque limité.

Cependant, SPIP "revient de loin" : son code, hérité du début des années 2000, aura mis bien du temps à être nettoyé (la version 1.9 a énormément apporté de ce côté là). Du coup, malgré une documentation claire, le noyau n'est pas très simple à modifier. Évidemment, un système de plugin très souple permet de contourner cette difficulté mais ils sont, eux, souvent mal documentés (<http://files.spip.org/spip-zone/>).

SPIP est un excellent "couteau suisse" de la publication : malgré quelques lourdeurs, il est extensible, souple et dynamiquement soutenu. Il ne rentre pas parfaitement dans le cahier des charges PLUME, mais à l'aide d'un certain nombre de plugins existants (et en lui en développant quelques uns), il devient possible d'en faire une plate-forme assez proche de celle demandée.

Points forts

- Facilité d'utilisation
- Chaîne de publication rapide et efficace
- Bonne extensibilité depuis la version 1.9
- Communauté francophone importante et réactive
- Bonne base d'installations dans la communauté Enseignement et Recherche

Points faibles

- Difficile de sortir du cadre de la publication d'articles (nombre de champs non configurables)
- Gestion du workflow limitée
- Gestion des ACL limitée (utilisation de plugins possible, mais ne correspondant pas aux besoins du projet PLUME).

2.10 - Textpattern

- Site officiel : <http://www.textpattern.com/>
- Grille CmsMatrix : [<http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=OqlzmI0dnvwShE2tEiOiQ>]
- Version évaluée : 4.0.4
- Documentation en ligne : <http://textbook.textpattern.net/>
- Support en ligne : <http://forum.textpattern.com/>
- Licence : GPL, bientôt BSD
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : moyenne, mais efficace
 - date de première / dernière release : 14/08/2005 (v4 - pas de v1,2,3) - 17/10/2006 (v4.0.4) cf Changelog
 - communauté francophone : plutôt restreinte par rapport à la communauté anglophone (voir [<http://forum.textpattern.com/viewforum.php?id=16>] ou [http://textbook.textpattern.net/wiki/index.php?title=Fr/Page_d%27accueil])
 - taille de la source : 1Mo
 - changelog : [http://textbook.textpattern.net/wiki/index.php?title=Textpattern_Release_History]
 - activité du développement : plutôt lente en ce qui concerne le noyau, relativement active en ce qui concerne les plugins

Informations complémentaires et avis dans le cadre du projet

A lire les forums et le changelog, Textpattern "semble" être un CMS en perte de vitesse. Sans pour autant être dépassé.

Il dispose de fonctionnalités intéressantes : système de workflow ([http://rpc.textpattern.com/help/?item=about_privileges&lang=fr limité]), gestion de plugins, templating simple et efficace. TextPattern semble en fait très proche de SPIP. Il y a cependant des différences de structurations. Par exemple là où SPIP joue sur Rubrique -> sous rubrique. -> sous

sous rubrique avec des mots clés associés , TextPattern fonctionne par Section (une et une seule par article) et par Catégorie (0,1 ou 2 par article). Des mots clés peuvent être ajoutés, mais sans pour autant permettre des manipulations avec comme le propose SPIP avec les squelettes.

Son emploi par des communautés francophones (et notamment la communauté enseignement sup et recherche) semble assez limité.

Points forts

- Processus de publication cohérent
- Interface claire
- Code concis et clair

Points faibles

- Logique parfois plus proche du blog que du CMS
- Limite du nombre de catégories
- Apparemment en perte de vitesse (activité de la communauté décroissante)

2.11 - Tiki CMS

- Site officiel : <http://tikiwiki.org/>
- Démo opensourceCMS :
[http://opensourcecms.com/index.php?option=com_content&task=view&id=180]
- Grille CmsMatrix : [<http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1025>]
- Version évaluée : 1.9.7
- Documentation en ligne : <http://ccurec01.in2p3.fr/~gosset/tikiwiki/>
- Support en ligne : <http://tikiwiki.org/forums>
- Licence : LGPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : relativement importante en anglais
 - date de première / dernière release : Octobre 2002 (v0.9) - 25/11/2006 (v1.9.7)
 - communauté francophone : très faible : http://tikiwiki.org/tiki-view_forum.php?forumId=7
 - taille de la source : 37Mo
 - changelog : <http://tikiwiki.org/tiki-index.php?page=changelog.txt>
 - activité du développement : active, mais très dispersée

Informations complémentaires et avis dans le cadre du projet

La courbe d'apprentissage semble particulièrement rude si l'on veut créer ses propres modules.

TikiWiki donne l'impression d'être une "boite à tout faire", mais cela se paie par une complexité globale importante (à titre d'exemple, Tikiwiki comprend dans son installation de base 194 tables SQL !) et des problèmes d'ergonomie (beaucoup trop d'options dans une interface confuse).

Cependant, il faut reconnaître que cette complexité permet à Tikiwiki d'avoir des fonctionnalités

plutôt pointues¹⁵, comme une gestion des ACLs et du workflow très fine.

Points forts

- Richesse des fonctionnalités
- Richesse des extensions
- Système de workflow très fin

Points faibles

- Interface confuse
- Courbe d'apprentissage plutôt longue pour une personnalisation

2.12 - Typo3

- Site officiel : <http://typo3.com/>
- Démo opensourceCMS :
[http://opensourcecms.com/index.php?option=com_content&task=view&id=181]
- Grille CmsMatrix : [<http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1023>]
- Version évaluée : 4.1.1
- Documentation en ligne : <http://typo3.org/documentation/>
- Support en ligne : <http://support.typo3.org:8080/jive/category.jspa?categoryID=4>
- Licence : GPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : importante
 - date de première / dernière release : 1998/2002 (betas) - 24/05/2002 (v3.0) - 02/04/2007 (v4.1.1)
 - communauté francophone : oui, active : <http://www.typo3.fr/> - <http://forum.typo3.fr/>
 - taille de la source : 20Mo
 - changelog : [<http://typo3.com/History.1268.0.html>]
 - activité du développement : forte

Informations complémentaires et avis dans le cadre du projet

Typo3 est, de loin, le plus complet (et le plus complexe !) des CMS évalués ici.

L'interface d'administration est particulièrement fournie, au prix d'une prise en main difficile et d'une ergonomie parfois peu cohérente : certains menus, liens ou options semblent avoir été ajouté là où il y restait de la place, plutôt que là où l'utilisateur s'attendrait à les trouver.

Nul doute qu'un expert Typo3 pourrait réaliser le cahier des charges PLUME en utilisant Typo3. Mais les experts Typo3 sont pour la plupart employés par des sociétés qui facturent justement ce type de prestation. Dans le cadre du projet PLUME, passer un mois à comprendre le fonctionnement d'un outil revient à courir le risque de faire une première version du site centré sur les fonctionnalités plutôt que sur l'interface utilisateur.

¹⁵ http://doc.tikiwiki.org/Features#Content_features

Points forts

- Grande extensibilité
- Système de gestion de plugins centralisé
- Fonctionnalités extrêmement riches
- Une communauté anglosaxonne active
- Un excellent support (notamment commercial)
- Produit qui a fait ses preuves dans tous types de contextes depuis plusieurs années
- Typoscript (langage puissant)

Points faibles

- Trop de fonctionnalités (!)
- Une courbe d'apprentissage très élevée pour ce type de produit (un mois d'autoapprentissage minimum recommandé sur le site officiel)
- Une ergonomie très moyenne
- Un système de thèmes puissant, mais vraiment lourd
- Typoscript (langage spécifique)

2.13 - Xaraya

- Site officiel : <http://www.xaraya.com/>
- Démo opensourceCMS :
[\[http://opensourcecms.com/index.php?option=com_content&task=view&id=173\]](http://opensourcecms.com/index.php?option=com_content&task=view&id=173)
- Grille CmsMatrix : [\[http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1026\]](http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1026)
- Version évaluée : 1.1.2
- Documentation en ligne : <http://www.xaraya.com/index.php/base/documentation>
- Support en ligne : <http://www.xaraya.com/index.php/xarbb/>
- Licence : GPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support : Moyenne et activité ralentie (beaucoup de messages non répondus)
 - date de première / dernière release : 27/10/2003 (v0.9.1.4, issue d'un fork de développeurs de postnuke) -
 - communauté francophone : inactive : <http://fr.xaraya.com>
 - taille de la source : 20Mo (base) - 50Mo (full)
 - changelog : ?
 - activité du développement : faible sur la branche 1.x

Informations complémentaires et avis dans le cadre du projet

- Le site officiel (et son pendant français) ont été inaccessibles a plusieurs reprises pendant les tests
- La communauté francophone semble à l'abandon
- Le développement/support de la branche 1.x semble ralenti
- L'application est réputée lourde et gourmande en ressources

Ce framework est issu du départ de développeurs du projet postnuke. L'outil semble très complet et puissant. La documentation est dense et de bonne qualité.

Comme tout framework, sa phase d'apprentissage pourrait s'avérer relativement longue. Cependant, le principal inconvénient de Xaraya me semble être son activité ralentie et les problèmes internes qui semblent agiter la communauté.

La lecture de la page <http://trac.xaraya.com/2x/wiki/XarayaNova> ne pousse en effet pas à retenir un outil dont le code pourrait être profondément revu lors de la prochaine version majeure et dont la direction de projet parait assez confuse :

We want to create a working environment independent of the accumulated routine activities created over time in the Xaraya project group

(...)

Proposed Solution : As our focus is on developing 2x, we want to ensure that the current problems the Xaraya project as a whole is experiencing won't hinder the progress of 2x. As a minimal solution we therefore propose creating a subproject within Xaraya specifically around 2x development. The subproject would operate in loose coordination with the rest of the Xaraya project.

(...)

The subproject will run autonomously within the Xaraya project.

Enfin, Xaraya donne l'impression que le développement de la branche 1.x serait quasiment arrêté et que l'activité de la communauté serait plutôt en récession par rapport à 2005/2006.

Points forts

- Bonne documentation
- Framework bien conçu
- Interface claire et agréable

Points faibles

- Situation actuelle et avenir du développement difficilement lisible
- Faible valeur ajoutée par rapport à d'autres frameworks concurrents
- Réécriture du code en cours pour une version 2

2.14 - Xoops

- Site officiel : <http://www.xoops.org>
- Démo opensourceCMS :
[http://opensourcecms.com/index.php?option=com_content&task=view&id=165]
- Grille CmsMatrix : [<http://www.cmsmatrix.org/matrix/cms-matrix?func=viewDetail;listingId=1051>]
- Version évaluée : 2.0.16
- Documentation en ligne : <http://xoopsdocs.net/modules/docs/>
- Support en ligne : <http://www.xoops.org/modules/newbb/>
- Licence : GPL
- Éléments (subjectifs) d'activité et de pérennité :
 - taille du support :
 - date de première / dernière release :
 - communauté francophone :
 - taille de la source :
 - changelog
 - activité du développement :

Informations complémentaires et avis dans le cadre du projet

Après étude de la démonstration et lecture de différents forums, j'ai écarté d'office ce CMS. Non pas qu'il ne puisse pas correspondre au cahier des charges PLUME, mais parce qu'en terme de fonctionnalités, il n'apporte rien de plus que ces concurrents directs (Joomla et Drupal) alors que son code, son ergonomie, sa pérennité et son développement semblent en retrait par rapport à ces derniers.