CETE de l'Ouest
centre d'Études techniques de l'Équipement

Gestion de contenu

Etude d'architecture du plugin « saisie facile »

C. Imberti – version du 04/03/2010 modifiée le 21/06/2010

Historique des versions du document

Version	Auteur	Commentaires
1	Ministère de l'Ecologie, de l'Energie, du Développement durable et de la Mer CETE de l'Ouest / DIOG / C. Imberti	Version du 04/03/2010
1.1	Ministère de l'Ecologie, de l'Energie, du Développement durable et de la Mer CETE de l'Ouest / DIOG / C. Imberti	Version du 21/06/2010 : ajout du chapitre 5.

Sommaire

1. Introduction

Extraits de l'étude visant à faciliter la création d'un article (CETE de l'Ouest / DIOG / C.Imberti – juillet 2008) :

La création d'un article doit être aussi simple que d'envoyer un message.

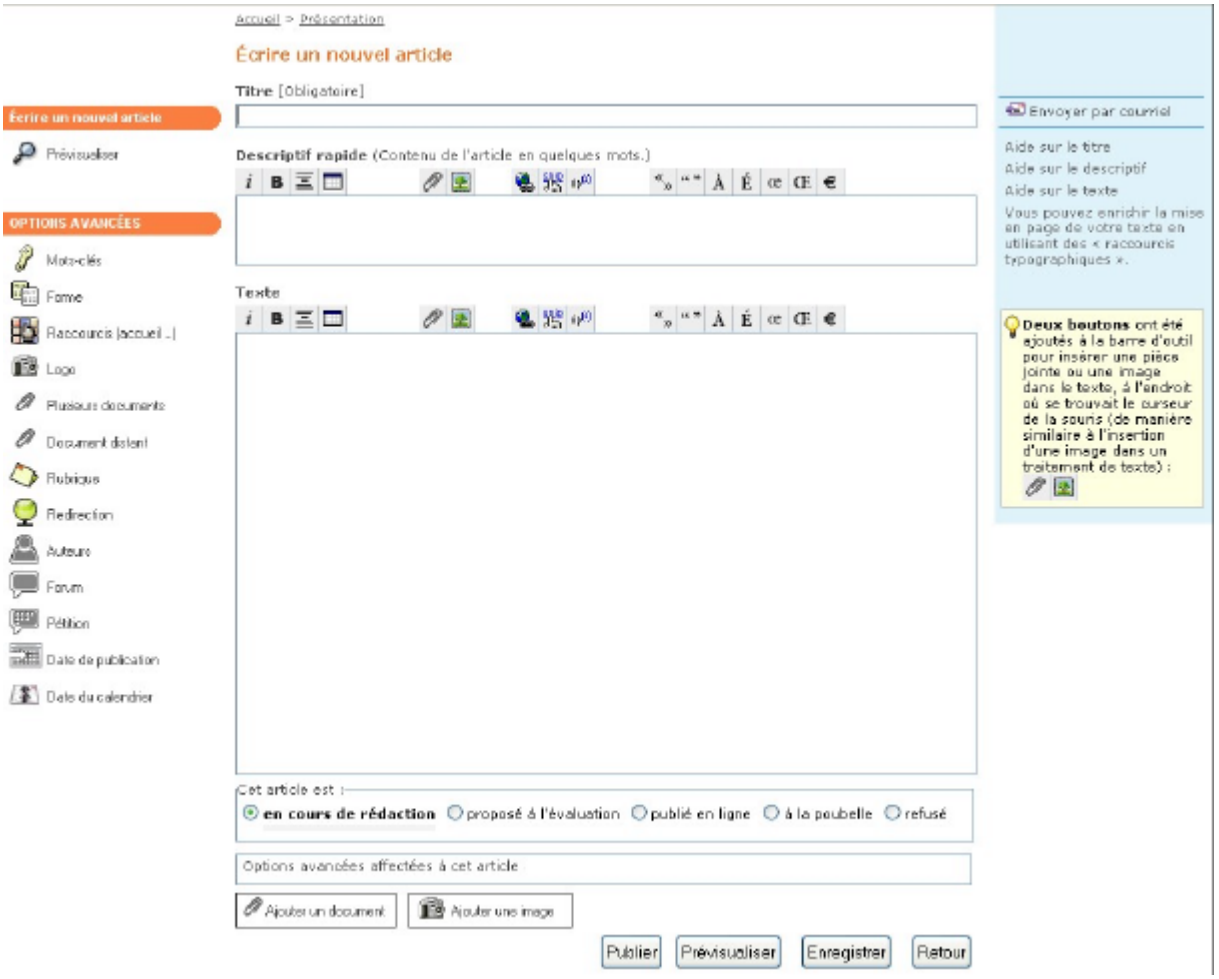
Les objectifs peuvent être détaillés de la manière suivante :

- Diminuer le nombre de clics nécessaires pour créer un article ;
- Eviter certaines complexités de SPIP (découpage en 2 pages, triangle pour déplier les blocs, …) ;.
- Simplifier l'insertion d'un document ou d'une image dans le texte ;
- Elargir le champ de saisie du descriptif et du texte ;
- Positionner automatiquement le curseur de la souris dans le champ « titre » ;
- Le bouton "modifier cette page", sur le site public, doit permettre d'arriver directement en mode "modification" de l'article ;
- Pouvoir accéder à l'ensemble des documents joints depuis les 2 pages (configuration et modification de l'article) ;
- Pouvoir remplacer une image en une seule manipulation, comme c'est le cas pour les documents ;
- Pouvoir créer un article dans le calendrier en cliquant sur une date du calendrier ;
- Offrir un niveau de sécurité identique à la solution actuelle ;
- Améliorer au passage l'accessibilité.

2. Situation existante (version 2.4 de Giseh sous SPIP 1.8.3)

2.1 Illustration en images

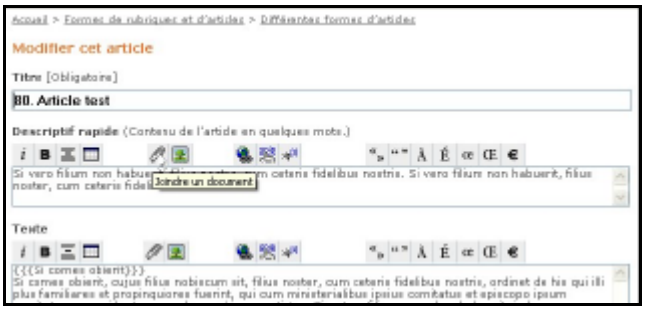
2.1.1 Formulaire de saisie d'un article

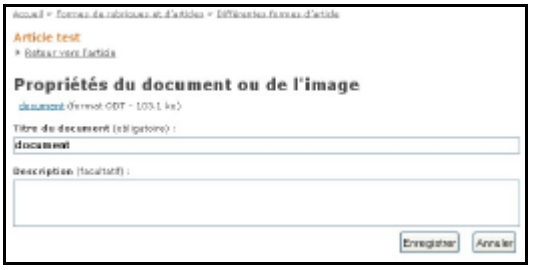


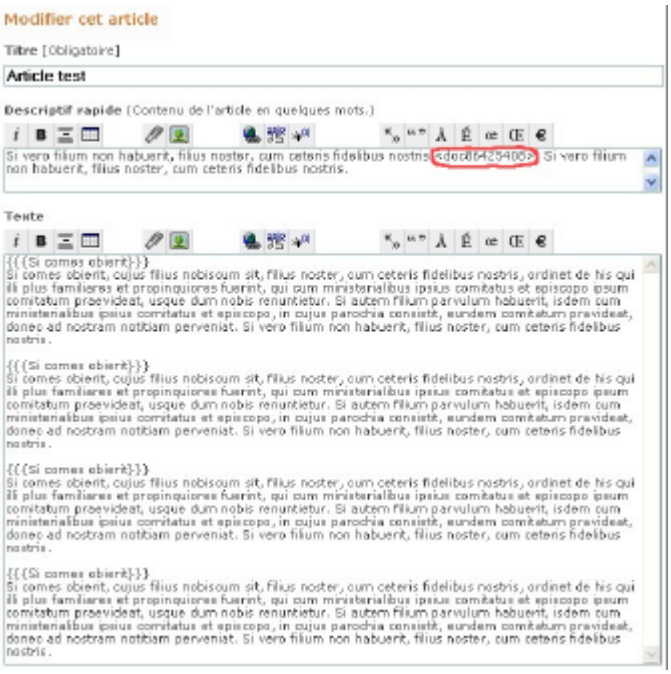
Remarques :
- si un auteur a le droit de publier un article et si le statut enregistré de cet article n'est pas «publié», alors un bouton «publier» figurera en bas du formulaire. L'action de ce bouton consistera à publier l'article et à le voir en ligne (ce qui évite certains clics).
- rappel, le cas échéant, du bouton «publier» ou «prévisualiser» en bas du formulaire, en fonction du statut enregistré pour l'article.
- lorsque l'on appuie sur «Entrée» dans le formulaire, cela active la première action (en haut de la colonne de gauche) à savoir prévisualisation ou voir en ligne.
- un rédacteur ne peut pas mettre le statut «en cours de rédaction» à un article qui est «proposé à l'évaluation» (règle de gestion de SPIP).
- L'article sera enregistré automatiquement quand on déclenche une action (même si javascript est désactivé sur le navigateur), afin d'éviter de perdre le texte saisi lorsque l'on ajoute une pièce jointe.
- La barre d'outil du texte et du descriptif apparaît uniquement si javascript est activé.

2.1.2 Ajout d'un document et insertion dans le texte

Deux boutons ont été ajoutés à la barre d'outil pour insérer une pièce jointe ou une image dans le texte, à l'endroit où se trouvait le curseur de la souris (de manière similaire à l'insertion d'une image dans un traitement de texte) :





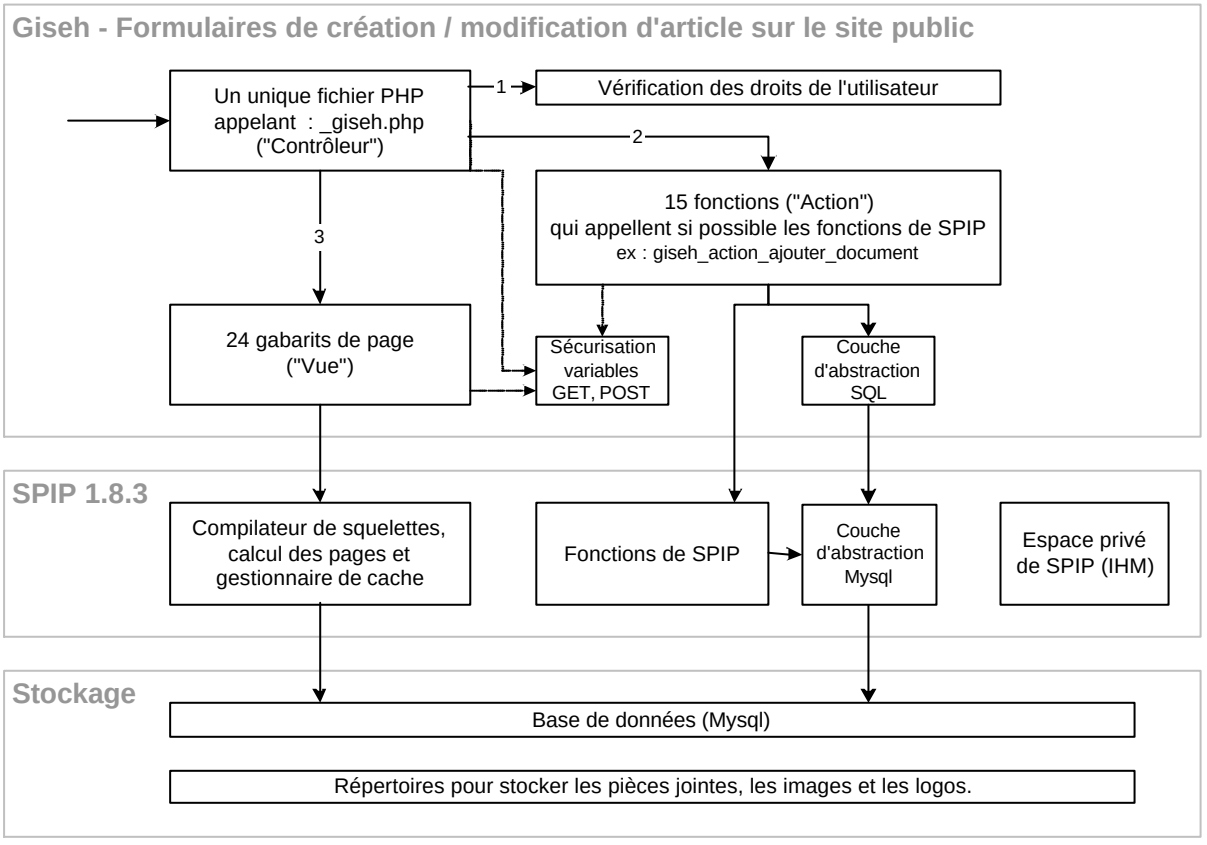


2.2 Architecture actuelle

Sur le plan technique, il convient d'éviter d'opérer sur le code de l'espace privé qui est largement remanié à chaque nouvelle version de SPIP. Par ailleurs, un squelette est, sauf exception, portable d'une version de SPIP à une autre. D'où l'intérêt de confier la partie «Vue » du modèle vue contrôleur (MVC) aux squelettes.
Extrait de http://www.spip.net/fr_article3497.html : « *ces fonctions devant être à terme les filtres utilisés par les fichiers de exec quand ils seront des squelettes* ».

Aussi, l'architecture actuelle se rapproche du modèle MVC, sans toutefois l'atteindre :

- un unique fichier PHP appelant joue le rôle du contrôleur (_giseh.php).
- des squelettes jouent le rôle de vue (_template/_giseh_….html).
- les actions et la partie modèle ne sont pas distinctes. Elles figurent dans le répertoire « ecrire/giseh_action ».



Un mécanisme de sauvegarde automatique de l'article est intégré dans l'unique fichier PHP appelant qui joue le rôle du contrôleur.
Une couche d'abstraction pour les requêtes SQL est utilisée. Les fonctions de SPIP sont largement utilisées (comme SPIP n'est pas codé en objet, les formulaires pour la « saisie facile » n'utilisent pas d'objet).

Cette architecture a été conçue en tenant compte des évolutions de SPIP.

3. Les nouveautés de SPIP 2.0 sur ce sujet

3.1 Un seul fichier d'appel pour les pages du site public (spip.php)

Extrait de http://www.spip.net/fr_article3497.html

SPIP comprend un seul fichier d'appel pour les pages du site public (spip.php alias index.php), chargeant le fichier d'initialisation ecrire/inc_version.php et passant immédiatement la main au script principal ecrire/public.php ;

Le fichier ecrire/public.php, appelé par spip.php, a pour rôle essentiel de délivrer les pages de l'espace public, commandées lorsque la requête HTTP comporte (après réécriture éventuelle) le paramètre page.

L'autre rôle de ecrire/public.php concerne le cas où la requête HTTP comporte l'argument action. Il applique alors la fonction charger_fonction à la valeur v de ce paramètre action. Cette application a pour effet de charger le fichier homonyme du répertoire action, dont la fonction principale action_v_dist est alors invoquée. Ces scripts effectuent essentiellement des écritures (en base ou sur fichier) et ne retournent en général pas de résultat.

3.2 La méthode CVT pour les formulaires (charger, vérifier, traiter)

Extrait de http://www.spip.net/fr_article3800.html

La réalisation de formulaires dynamiques a été simplifiée avec SPIP 2.0 grâce à un formalisme clair découpé en une vue (ou squelette) pour l'affichage, et trois étapes Charger, Vérifier, Traiter (CVT).

Prenons l'exemple d'un formulaire de contact que l'on veut gérer avec la balise #FORMULAIRE_CONTACT.

Affichage
Lorsque SPIP rencontre la balise #FORMULAIRE_CONTACT, il reconnaît qu'il s'agit d'une balise du type #FORMULAIRE_xxx. Il cherche alors le squelette formulaires/contact.html pour afficher le formulaire. Il n'y a aucun autre pré-requis pour l'affichage du formulaire, ce qui permet son intégration visuelle indépendamment des 3 fonctions ci-dessous.

Chargement
Avant l'affichage du squelette, SPIP appelle, si elle existe, la fonction formulaires_contact_charger_dist() pour fournir la liste des champs saisis dans le formulaire, avec des valeurs par défaut éventuelles. Ces champs et valeurs seront fournis au squelette formulaires/contact.html qui en fait donc usage.

Vérification
Lorsque l'internaute remplit le formulaire et clique sur le bouton de validation, SPIP appelle la fonction formulaires_contact_verifier_dist() pour vérifier la validité de la saisie. La fonction renvoie une liste de messages d'erreur correspondants à chaque champs erroné, ou une liste vide en cas d'absence d'erreur.

Traitement
Si la fonction de vérification n'a pas renvoyé d'erreur, alors SPIP appelle automatiquement la fonction de traitement du formulaire formulaires_contact_traiter_dist() qui pourra réaliser toutes les opérations de traitement du formulaire : envoi d'un mail, enregistrement en base de donnée, etc.

3.3 Une factorisation du contrôle des droits d'accès et de modifications

Une balise #AUTORISER permet de demander des autorisations dans un squelette. La présence de cette balise crée un cache de squelette par visiteur identifié et un seul cache pour les visiteurs non identifiés.
Exemple :

```
[(#AUTORISER{modifier,article,#ID_ARTICLE})
        ... actions
]
```

3.4 Une couche d'abstraction de données

SPIP 2.0 peut lire, écrire et fonctionner à partir de gestionnaires de bases de données MySQL, PostGres et SQLite. Bien que leur syntaxe d'écriture des requêtes soit différente, SPIP, grâce à un jeu de fonctions spécifiques d'abstraction SQL permet de coder des interactions avec la base de données indépendantes de celles-ci.

4. Une nouvelle architecture dans le cadre de SPIP 2.0

4.1 Utiliser le contrôleur de SPIP (spip.php) et pas celui de la nouvelle interface (_giseh.php)

Le mécanisme de sauvegarde automatique de l'article ne peut plus être intégré dans le fichier PHP d'appel qui joue le rôle du contrôleur, puisque l'on utilise désormais le fichier d'appel de SPIP.

La solution consiste à reprendre tout le mécanisme de sauvegarde et d'aiguillage effectué par le fichier _giseh.php et à le placer dans la fonction de traitement associée au formulaire de création / modification d'article.

A noter que pour que le mécanisme de sauvegarde automatique fonctionne, il faut que toute action depuis le formulaire de création / modification d'article valide celui-ci, avant d'être aiguillée le cas échéant vers un formulaire secondaire. Ceci nécessite d'utiliser des balise HTML «input » au lieu de simples liens. En effet, pour des questions d'accessibilité, la sauvegarde automatique doit fonctionner même si javascript est désactivé. Pour éviter des traitements inutiles, la sauvegarde automatique doit être effectuée uniquement si le titre, le descriptif ou le texte de l'article a évolué.

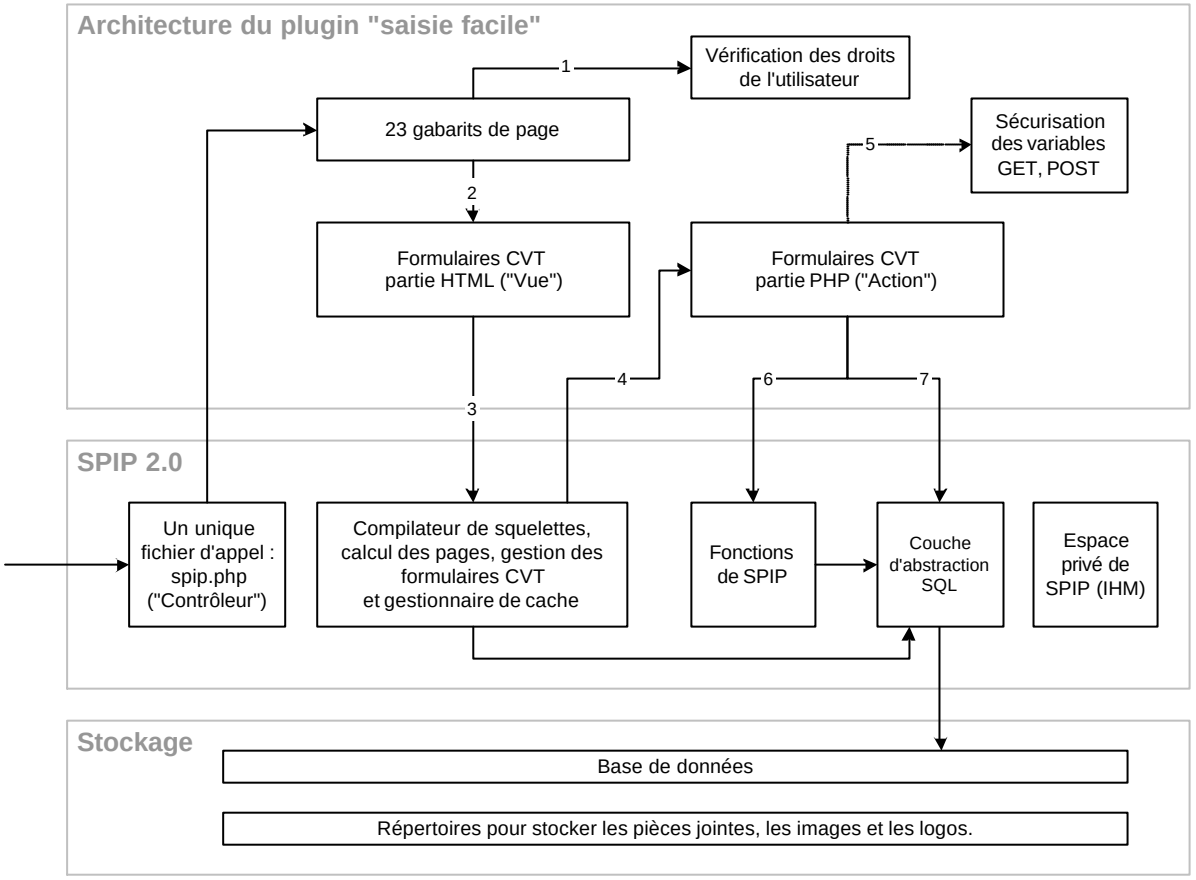
4.2 Utiliser les fonctions de SPIP de contrôle des droits d'accès et de modifications

Il convient de s'appuyer sur ces fonctions centralisées. Le cas du statut de visiteur doit être examiné avec précaution car comme il n'a pas accès à l'espace privé, les fonctions centralisées de contrôle des droits d'accès n'en tiennent pas forcément compte.

Une solution consiste à interdire l'accès aux formulaires à toute personne non authentifiée ainsi qu'aux utilisateurs qui ont le statut de « visiteur ». Ceci permet d'être complètement aligné sur les mécanismes de sécurité de SPIP 2.

4.3 Utiliser l'architecture CVT de SPIP

Il convient de d'extraire la partie formulaire de la page actuelle de création / modification d'article et de découper ce formulaire en un fichier HTML et un fichier PHP contenant les traitements CVT.



A noter que le fichier HTML ne doit plus contenir de code PHP (grâce aux nouvelles possibilités de SPIP 2.0 : #SET, #GET et filtres personnalisés).

Pour le traitement de l'article, SPIP fournit la fonction « formulaires_editer_objet_traiter », toutefois il est préférable de créer une fonction spécifique pour le traitement des formulaires secondaires, car la fonction de SPIP précitée génère un message dans les logs de SPIP indiquant qu'une table n'a pas été trouvée.

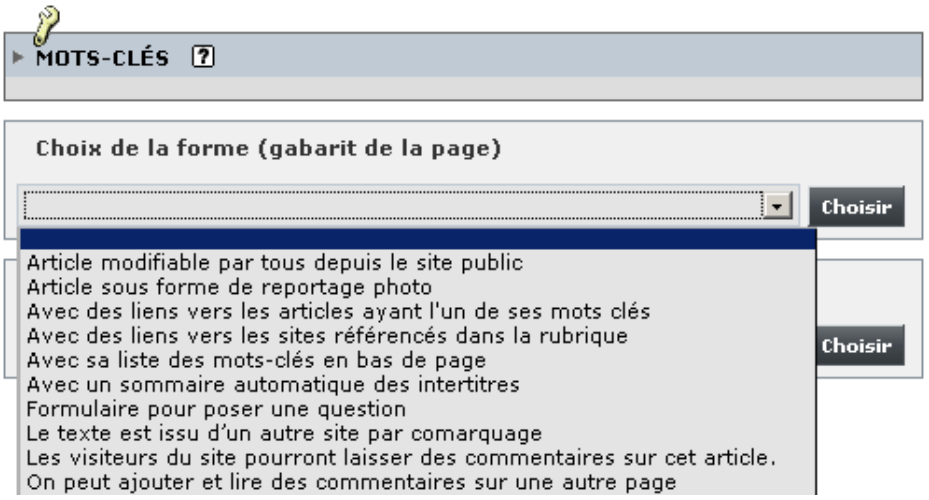
Le principe de l'architecture évolue peu car elle avait été conçue dès le départ en tenant compte des évolutions de SPIP.

En revanche, l'adaptation au modèle CVT nécessite de modifier la vingtaine de formulaires.

4.4 Offrir des pipelines

4.4.1 Options avancées additionnelles

Grâce à un pipeline offert par SPIP, le plugin « ciparam : configurateur de squelettes » peut afficher dans la page de saisie de l'article, dans l'espace privé, un sélecteur de forme d'article et un sélecteur de raccourcis.



Aussi, l'interface de saisie simplifiée sur le site public doit offrir au plugin « ciparam : configurateur de squelettes » une possibilité équivalente.

La liste des options avancées devra pouvoir être enrichie via un pipeline.

Corrélativement, la zone, de l'interface de saisie simplifiée, qui détaille les options avancées affectées à l'article devra également pouvoir être enrichie via un pipeline.

Les formulaires secondaires correspondants à ces options supplémentaires, ajoutées via un pipeline, figureront dans le plugin qui utilise le pipeline. Par exemple, le formulaire concernant les formes d'articles figurera dans le plugin « ciparam : configurateur de squelettes ». C'est logique, vu que ce dernier contient déjà le sélecteur de forme d'article qui s'affiche dans l'espace privé. C'est nécessaire, en termes d'indépendance des plugins. Les fichiers contenant ces formulaires devront avoir un nom qui commence par « cisf_ », non seulement pour les reconnaître, mais aussi pour faciliter une éventuelle interdiction d'accès à ces pages depuis internet (au niveau de Apache).

4.4.2 Outils additionnels

L'interface de saisie simplifiée sur le site public, comprend en haut de la colonne de droite un bloc d'outil avec le lien « Envoyer par courriel », etc. Par exemple, dans le cadre des accès restreints, il convient d'ajouter un lien « Envoyer aux membres ».

Aussi, la liste des outils devra pouvoir être enrichie via un pipeline.

Si un outil additionnel appelle une page, cette dernière devra figurer dans le plugin qui utilise le pipeline. Cette page devra avoir un nom qui commence par « cisf_ »,

4.5 Volumétrie des rubriques

La solution consiste à se baser sur le filtre « chercher_rubrique » de SPIP 2, dans le formulaire secondaire de changement de rubrique. Il convient d'appliquer l'ergonomie qui existe sous Giseh v2.4 sur ce point.

4.6 Indépendance vis-à-vis des chartes graphiques

Le plugin peut être utilisé sur des sites qui ont des chartes graphiques diverses et variées (nombre de colonnes, etc.). Aussi, le plugin devra être indépendant des chartes graphiques. Pour ne pas dérouter l'utilisateur, il utilisera des icônes et des couleurs de SPIP.

4.7 Compatibilité avec un plugin d'éditeur WYSYWYG (plugin ckeditor)

Le plugin ckeditor est l'évolution du plugin fckeditor (du même auteur).
Il n'est pas compatible avec le plugin « cisf ». En effet, dans son pipeline, il teste la valeur de $_GET['exec'], pour savoir si on se trouve dans la page d'édition de l'article dans l'espace privé.
Ceci empêche donc son utilisation depuis le site public. Toutefois, une solution consiste à le leurrer en déclarant dans le formulaire de saisie d'article sur le site public : $_GET['exec']="articles_edit";

Ckeditor nécessite d'inclure un fichier javascript de Jquery dans le formulaire de saisie d'article. Vu la taille de ce fichier javascript, il convient l'inclure uniquement si le plugin ckeditor est actif.

Dans ckeditor, des chemins «../ » sont codés en dur, ce qui empêche son utilisation depuis le site public. J'ai posté un commentaire sur le forum de ce plugin en indiquant qu'il conviendrait d'utiliser la constante _DIR_RACINE à la place. Pour l'instant, il n'a pas été modifié.

Remarque : quand on passe de l'éditeur SPIP à l'éditeur WYSIWYG on perd certaines mise en forme de SPIP.

4.8 Cas spécifiques

4.8.1 Formulaires multi enregistrements

Concernant les formulaires secondaires, comme par exemple celui permettant d'affecter des auteurs à l'article, il s'agit de trouver une approche pour les formulaires multi enregistrements.
Une solution consiste, dans la fonction de chargement du formulaire, à passer, dans la liste des valeurs, un tableau des identifiants existants (par exemple, ceux des auteurs déjà affectés à l'article) puis de les utiliser dans la page HTML à travers la balise #ENV.
Cela ne suffit pas, car la la fonction de SPIP formulaires_editer_objet_traiter n'est pas adaptée aux formulaires multi enregistrements. Aussi, il faut créer un fichier « action » spécifique avec les traitements corrélatifs.

4.8.2 Upload de document

Afin de bénéficier des services offerts par SPIP , la solution consiste à créer une balise (#SF_JOINDRE) qui fait appel à la fonction inc/joindre.php de SPIP.

A noter que l'option avancée « document distant » n'est plus proposée car elle est désormais intégrée dans l'upload de document.

4.8.3 Supprimer un document

Il convient d'utiliser le mécanisme de suppression de document de SPIP. Une des conséquences est que, désormais, la suppression est directe comme dans SPIP (sans passer par une phase de confirmation).

4.8.4 Boutons d'administration

Une solution consiste à utiliser un formulaire administration.html personnalisé pour afficher les liens vers la modification ou création d'article.
Pour les rédacteurs, il convient de poser automatiquement le «cookie de correspondance » (au sens SPIP) afin de leur afficher les boutons d'administration.

5. Bug de SPIP très gênant pour la saisie (avec le mode révision activé)

Lorsque le suivi des révisions est activé, parfois l'enregistrement d'une modification de l'article prend une dizaine de secondes.

J'ai examiné la fonction ajouter_version (dans inc/revisions.php) et j'ai ajouté des traces dans les logs :

a) par exemple, la fonction insère pour l'article 485 une version (-88160935208126) avec dans "titre_version" : 1276758535.20812500.

b) avant la boucle "while (sql_countsel ", j'ai ajouté des traces (spip_log) pour obtenir le code de la clause where de la première requête MYSQL de la boucle :
id_article=485
AND id_version < 0
AND 0.0+titre_version < 1276758535.20812500
AND 0.0+titre_version > 1276758525

j'ai également ajouté des traces pour obtenir le résultat du select correspondant à cette clause where :
SELECT * FROM spip_versions WHERE id_article =485
AND id_version <0
AND 0.0 + titre_version < 1276758535.20812500
AND 0.0 + titre_version > 1276758525

Une ligne trouvée :
id_article : 485
id_version : -88160935208126
titre_version : 1276758535.20812500

Le select trouve une ligne avec "titre_version" égal à 1276758535.20812500
alors que la clause WHERE comprend 0.0 + titre_version < 1276758535.20812500

Ce résultat est surprenant.

Avec phpmyadmin, j'ai reproduit le phénomène et j'ai ajouté 0.0+titre_version dans le champ du select:
titre_version : 1276758535.20812500
0.0+titre_version : 1276758535.2081

Visiblement, MYSQL limite « 0.0+titre_version » à quatre décimales.

Aussi, dans la fonction ajouter_version, j'ai remplacé
$date = $sec . substr($ms,1);
par
$date = $sec . substr($ms,1,4);

et depuis, le problème ne s'est pas reproduit.

J’ai communiqué cette analyse à la communauté SPIP le 17 juin 2010 à 09h55. A 10h06, je recevais la réponse suivante de la communauté SPIP :

From: cedric…
Subject: Re: Probleme dans la fonction ajouter_version
Newsgroups: gmane.comp.web.spip.devel
Date: 2010-06-17 10:06:01 GMT (4 minutes ago)

Magnifique !

http://trac.rezo.net/trac/spip/changeset/15778

Ce bug traîne depuis le début et personne n'avait encore eu le courage de le débusquer.

Cédric

Comme cela figure dans le message, la communauté a apporté la correction sur le code de la version 2.1.n. Toutefois, il faudra attendre la sortie de la prochaine version 2.1.n pour en profiter.

En attendant, une solution consiste à désactiver le mode révision.