

Libroscope, plus que des
logiciels libres, des hommes
libres !

-- Documents - Conférences --

Conférences

RMLL 2003 - METZ 9-12 Juillet

Spip au scope.

Retranscription de la
conférence d'Antoine Pitrou.

Thierry Pinon
samedi 14 février 2004

Dans le cadre des Rencontres Mondiales du Logiciel Libre dont l'édition 2003 s'est déroulée à Metz du 9 au 12 Juillet 2003, Antoine Pitrou a présenté une conférence traitant des aspects organisationnels et méthodologiques du projet SPIP, logiciel dont il est co-auteur.

Cet article retranscrit les propos tenus par Antoine. Vous trouverez les transparents et l'enregistrement audiophonique de cette conférence sur

<http://www.libroscope.org/doc/conf/...>

Préambule SPIP est un système de publication pour internet francophone, ce qu'on appelle aussi parfois un CMS, [Content Management System](#). Dans le cas de SPIP, il s'agit d'un ensemble de script [PHP](#) que l'on installe chez son hébergeur, on configure l'ensemble, on arrive sur une interface de rédaction qui permet de créer des rubriques, des articles et les mettre en ligne de façon simple, avec une interface intuitive et une gestion de raccourcis typographiques. Un workflow de validation d'articles permet à une équipe d'administrateurs de gérer le site et la ligne éditoriale. L'idée générale de SPIP est l'auto publication sur le web pour tous, sans connaissances techniques. SPIP a été créé au départ par les administrateurs d'[Uzine](#) qui est un site qui milite pour la liberté d'expression sur le web. SPIP est un outil qui permet de mettre en oeuvre cette liberté d'expression de façon très simple et efficace. Sa principale caractéristique extra-technique est qu'il est populaire, assez apprécié en général, et utilisé par une population diversifiée, aussi bien pour les sites personnels que pour les associations ([Réseau Voltaire](#), [Reporters sans Frontières](#) ou par des journaux comme [Le Monde Diplomatique](#) ou [l'Humanité](#)).

SPIP : l'outil

Historique. Le créateur originel de SPIP est ARNO* pour le site Uzine2. A l'origine, Uzine était un site collaboratif géré manuellement, avec un accès [ftp](#) partagé par l'ensemble des auteurs, qui codaient leurs documents [html](#) à la main, ce qui n'était pas très pratique et nécessitait des connaissances techniques. ARNO* en 2000 a décidé d'avoir un vrai système pour Uzine2 qui permet de publier des articles sans avoir de code ftp et de connaissances techniques. Ensuite sont arrivés successivement dans l'équipe de développeurs, Antoine (administrateur d'Uzine) et de Fil (journaliste et webmestre du Monde Diplomatique). La première version officielle de SPIP est sortie en juillet 2001, après une gestation d'un an. Depuis, sa popularité est en croissance régulière, c'est à dire que fin 2001 il y avait 2000 téléchargements par mois et aujourd'hui (ndlr : Juillet 2003) entre 10000 et 15000. Régulièrement, l'utilisation de SPIP a augmenté au fil des sortis et grâce au bouche à oreille.

Besoins. SPIP a été créé pour résoudre des problèmes de « webmastering », c'est une application métier pour employer un terme du « jargon entreprise ». La conception de SPIP s'articule sur trois niveaux : les aspects opérationnels, fonctionnels et techniques.

Les aspects opérationnels définissent plus ou moins l'esprit SPIP, c'est à dire travailler à plusieurs sans se taper sur les doigts : faire travailler plusieurs rédacteurs responsables de leurs articles respectifs, saisir les textes de manière naturelle (pas de balises html par exemple...), installer l'outil en un clin d'oeil (pré-requis pour la diffusion de l'outil), structurer son site sans mal de crane (gestion de la ligne éditoriale, de la navigation, de la structuration des rubriques), modifier aisément la mise en page par un système de templates (gabarit en français, squelettes en SPIP) qui permet, par exemple, de modifier une description de mise en page qui va permettre de changer la mise en page de tous les articles.

Les problématiques fonctionnelles sont la mise en oeuvre conceptuelle des problématiques opérationnelles : créer ces articles dans une structure fine, c'est à dire une arborescence de rubriques (choix finalement assez naturel), permettre aux visiteurs d'intervenir sur le site pour le faire vivre à travers les forums et les pétitions, présenter joliment les articles via les raccourcis de mise en page (intertitre, structurer l'article etc...), communiquer avec d'autres sites à travers la syndication (échange d'articles entre sites).

Enfin, **la problématique technique** du point vu du programme proprement dit. Première contrainte par exemple, afficher les pages rapidement, ne pas surcharger le serveur (beaucoup de sites sont hébergés sur des serveurs mutualisés). Maximiser la compatibilité avec les différents hébergeurs (plus petit dénominateur commun, par exemple le choix du php3) afin de faciliter l'utilisation de SPIP quel que soit l'hébergeur. L'exigence sur laquelle il faut insister en ce qui concerne SPIP est l'accessibilité au plus grand nombre au sens général, c'est à dire utilisable par tous : des raccourcis facilement assimilables, une installation automatique par le web de type « Wizard » (il n'y a plus qu'à cliquer...), une gestion transparente de la base de données. De nombreuses fonctionnalités de SPIP sont apparues au fil du temps pour satisfaire cette exigence d'accessibilité.

Insuffisances. Vis à vis de ce souci d'accessibilité, on peut repérer quelques insuffisances liées au souci d'utilisabilité justement : on évacue de fait la partie customisation (personnalisation) et automatisation à outrance. Par exemple, pour changer de rubrique quinze articles, vous allez devoir faire quinze fois la même manipulation, ce qui n'est pas adapté à l'administration de masse. Il subsiste quelques « uzinismes » (du nom du site) qui sont des fonctionnalités développées à l'origine pour le site Uzine et pas très utilisées, les pétitions par exemple ou le calendrier interne, la messagerie interne et les mots clés sur les forums. Il est difficile de marier facilité d'utilisation et souplesse d'usage, il faut faire un choix, et du côté de SPIP ce choix est clair, c'est la facilité d'utilisation qui prime, l'utilisabilité au jour le jour jusqu'à l'ultra débutant. Et ce, au détriment de l'ultra-configurabilité.

La « communauté » SPIP

Communauté entre guillemets, car on ne sait pas précisément ce que cela représente, il faut donc être prudent.

Les outils de communication de SPIP.

Au niveau du **web**, c'est tout d'abord un site en français avec spip.org qui est très complet, il reprend l'aide en ligne, comporte surtout un très gros chapitre sur la mise en page du site public qui traite du fonctionnement des squelettes qui est la partie la plus délicate de SPIP. Il existe également un site de contributions externes, [Spip-contrib](http://spip-contrib) qui permet d'échanger des squelettes par exemple, le site de téléchargement, un archivage des discussions, un site de communication multilingue spip.net qui pour l'instant est en travaux et sur lequel il n'y a pour l'instant pas grand-chose car l'internationalisation de spip est récente et prend du temps (ndlr : depuis, les choses ont évoluées et spip.net est devenu le site officiel de spip et s'est nettement enrichie).

Au niveau des **listes de discussions**, on retrouve classiquement une [liste des annonces](#) (pour les nouvelles versions). La [liste de discussion utilisateurs](#) de spip, c'est la principale, celle qui est la plus

utilisée et qui structure le noyau de la communauté d'utilisateurs spip. [La liste de discussion des développements](#), et non pas des développeurs, car des développeurs il y en a très peu, des gens qui suivent et discutent des développements, il y en a beaucoup ;-). On a également l'envoi automatique des [changements du CVS](#), outil qui a été intégré le jour où l'on est passé sous CVS. Les listes citées précédemment sont francophones, mais il existe également des [listes non-francophones](#) (en anglais, allemand...) qui pour le moment réunissent peu de monde car SPIP international est en voie de décollage.

La documentation web, [spip.net](#), est elle-même réalisée sous SPIP, ce qui valide le concept, prouve que ça marche et que ça réalise un site qui est navigable, qui est utile et qui fonctionne bien. Une des principales caractéristique de la documentation est que l'on privilégie ce qui est clarté et accessibilité, pas mal d'articles pour débutants, des [FAQs](#) (Questions fréquemment posées), une certaine redondance sur certains points, de préférence à l'exhaustivité, c'est à dire qu'il y a beaucoup de micro-fonctionnalités très techniques que l'on ne retrouve pas forcément dans la documentation mais sur lesquelles on se fera un plaisir de répondre aux gens sur les listes de discussions, si on nous pose la question.

Au chapitre des problèmes de sites, il y a des problèmes de mises à jour, la documentation spip est rattachée à Uzine, ce qui est motivé par le rattachement historique de SPIP à Uzine, mais qui n'est pas très commode puisque les administrateurs d'Uzine ne sont pas les développeurs de SPIP actuels, ce qui limite un peu les possibilités de mise à jour, d'affinement des articles, d'ajout de nouveaux articles. Des forums d'aide ont été créés sur ce site et sont redondants par rapport aux listes de discussions et moins bien suivis.

La liste des utilisateurs.

C'est le médium principal, elle est non modérée en écriture et en inscription, elle est francophone et regroupe 600 personnes, génère environ 50 messages par jour. Différents types de profil techniques se rencontrent sur la liste. Il y a le débutant total qui veut faire son site et qui a entendu parler de SPIP dans des journaux et qui pose des questions sur le logiciel à utiliser pour transférer des fichiers sur le FTP. Ils obtiennent tout de même des réponses à leurs questions, il y a très peu de gens qui font des réponses du style "RTFM". Il y a le webmestre amateur semi-pro, des gens déjà qualifiés en webmastering ou html et qui ont par exemple un site d'association, donc qui savent un peu faire et qui ont choisit SPIP car il se trouve que l'outil leur facilite la tâche. Ces gens là posent en général des questions liées à la customisation de la mise en page. Il y a les professionnels de l'informatique car SPIP commence à être utilisé en entreprise, des consultants qui demandent un peu tout et n'importe quoi et à qui il faut répondre tout de suite ce qui est un peu pénible. On a finalement peu de gens qui utilisent SPIP, de gens qui saisissent des articles sans accès au FTP, ce qui prouve par défaut que pour ce genre de tâches le logiciel est assez convivial et qu'il est suffisant pour tout ce qui est interactions éditoriales.

On trouve sur cette liste **différents profils** idéologiques ; beaucoup de militants sont venus à SPIP par le bouche à oreille, on retrouve ainsi des sites de gauche, des syndicats, Reporters Sans Frontières, [Vacarme](#), Le Monde Diplomatique, l'Humanité etc.. Sinon, le profil dilettante, le webmaster qui fait son site avec une bande de copain, du style weblog ou portfolio en ligne, ou site sur les bonzaï nain, ce genre de choses. Et enfin, le pragmatique qui bouffe à tous les râteliers et qui cherche à vendre son truc aux entreprises. Sinon, il y a quelques [? étoiles filantes ?](#), souvent des gens bien intentionnés qui lancent des trolls sur, par exemple, les brevets logiciels...il s'en suit une enfilade de 15 messages sur le sujet ce qui n'est pas forcément très pertinent sur une

liste SPIP. Il y a également les pontificateurs en normes [W3C](#), c'est une longue histoire, on pourrait faire une présentation entière sur le sujet. En effet, SPIP historiquement ne respecte pas beaucoup les standards du W3C et régulièrement il y a des gens qui viennent non seulement dire qu'il faut respecter les standards, mais ils viennent surtout dire que les standards c'est génial, ils expliquent pourquoi en recopiant la documentation W3C presque texto ou alors ils paraphrasent... Et puis il y a aussi les pontificateurs en méthode de programmation, de temps en temps il y a quelqu'un qui nous dit que [Java](#) c'est mieux, que l'orienté objet c'est bien et qu'il faut réécrire SPIP, et qu'il est prêt à nous donner des conseils en architecture...

Pour parler un peu plus sérieusement, le **thème principal**, comme je disais, c'est l'écriture des squelettes, les boucles qui sont les structures de base du langage de templates SPIP. C'est un petit langage de programmation qui est très simple et très déclaratif et qui nécessite quand même, notamment quand on est débutant et qu'on a jamais touché un langage de programmation, un petit peu de persévérance. Il y a les fonctions avancées qui ne sont pas très documentées et qui sont sujets de discussions. Il y a des questions annexes qui sont un peu hors sujet, mais qui ont quand même leur place sur la liste comme la configuration apache, les fichiers .htaccess etc...

L'ambiance de la liste est assez agréable, très bon enfant, on n'a pas l'impression de lire un fil de questions, mais quelque chose qui vit. Les professionnels sont souvent plus secs, genre ["j'ai promis à mon boss de lui livrer une démo, il faut que ça soit prêt pour 23h00, aidez-moi"](#). Une chose à souligner, c'est que les développeurs de SPIP sont assez présents sur la liste des utilisateurs, on l'est moins actuellement, on l'était surtout au début car il y avait très peu de gens qui connaissaient l'outil ; ce qui permet maintenant au gens qui ont appréhendé l'outil grâce à nos conseils de jouer notre rôle sur la liste et d'aider les nouveaux venus. Il y a un travail d'accompagnement qui n'est pas négligeable du tout.

L'évolution des utilisateurs. Au départ, on avait beaucoup de militants qui avaient été mis au courant de SPIP par le bouche à oreille ou encore un public de ["lurkers"](#) qui s'étaient dit ["ça va être bien, il faut que je sois là pour le télécharger quand il sera disponible"](#). Par la suite, on a eu des utilisateurs précoces qui sont souvent les plus enthousiastes, passionnés par la philosophie SPIP ; Au point parfois de rentrer en conflit ouvert avec les développeurs SPIP, des gens qui ont tellement fait leur cette philosophie, que cela entraîne des déceptions, allant jusqu'à (c'est arrivé une fois) nous qualifier de traîtres et dire que l'on allait détruire la philosophie SPIP...c'est assez étonnant. Au final, la liste c'est assagie suite à la banalisation de SPIP.

Si on fait le **bilan**, on peut dire que l'entraide fonctionne bien, les problèmes sont résolus assez rapidement ; les débutants pris en main, ce qui va bien avec la philosophie SPIP, il serait dommage qu'il en soit autrement. La liste à quelque peu perdu en chaleur, elle s'est normalisée par rapport à d'autres listes logiciels libres. Malgré certaines craintes, la liste n'a pas trop explosé en trafic, il est relativement stable. Il y a juste une question pour moi, est-ce qu'il y a un risque de fonctionnement trop routinier qui transformerait cette liste en forum technique tout bête en perdant cet esprit de communauté. C'est une question ouverte, on peut difficilement savoir à l'avance ce que ça va donner.

SPIP : le développement.

Une petite description rapide du mode développement. D'un point de vu **logistique**, on est passé de un à trois développeurs (dans l'ordre Arno*, Antoine puis Fil). Coté travail de groupe, on a longtemps travaillé en FTP privé ce qui peut paraître étonnant, mais aucun de nous trois avions l'habitude de travailler en CVS, dont l'un des trois qui était pratiquement allergique à l'idée de travailler en CVS, un

autre qui était neutre. En ce qui me concerne, j'avais déjà essayé (RCS, Source Safe), à l'époque, je n'avais pas été très enchanté. Le CVS a pris du temps à arriver. Par le FTP privé, on se prévenait ; Attention, j'ai mis une nouvelle version, il faut prendre les nouveaux fichiers ;. Au final, on se marchait pas mal sur les pieds, on a finalement décidé que CVS remplissait sa tâche, sans être miraculeux, ça rend service. Le système de gestion de bugs qui n'est malheureusement pas mis à profit qui est Mantis, c'est un bon système, pas usine à gaz comme bugzilla, mais pas très médiatisé. Mais son peu d'utilisation est peut-être lié à la population de la liste pas habituée à ce genre d'outils. On a une liste de discussion qui réunit 300 personnes sachant qu'il y a peu de discussions hors liste sur le développement.

L'équipe. Arno* alias Arnaud Martin, créateur de Spip pour la gestion du site uzine, de métier et de compétences est webmestre, graphiste, ; PAOiste ; au sens général. En ce qui me concerne, je suis informaticien. J'ai rejoint spip car j'étais un des administrateurs d'uzine, que l'outil était assez prometteur du point de vu conceptuel et même pratique puisqu'il servait déjà, donc, j'ai décidé d'y participer. Et Philippe Rivière alias Fil qui est journaliste au Monde Diplomatique, qui est plus ou moins technicien (il s'occupe de la machine ce genre de choses) et qui lui s'occupait du site du ; diplo ; qui avant spip était un ensemble de scripts perl complètement ; bordéliques ; qui prenaient des bouts de fichiers textes et qui les mettaient ensemble pour générer des pages html ; c'était un machin totalement fait à la main et qui devenait non maintenable. Il suivait l'évolution de spip depuis quelques temps, et à l'instant où cela a commencé à devenir vraiment intéressant, il a décidé de l'utiliser pour le site du Monde Diplomatique et à partir de là, il est devenu développeur.

Un développement chaotique. La principale caractéristique du développement, qui a tendance à frapper les personnes qui arrivent sur la liste, est qu'il est assez chaotique. Il n'y a pas de calendrier prévisionnel des ; features list ;, pas de méthodologie stricte pour savoir quand on sort une version et ce que veut dire une version de spip, mais une ; todo ; indicative à rallonge, il y a de quoi faire. Ce n'est pas un défaut de fonctionnalité, mais plutôt un excès de possibilités. Temporellement, le développement se déroule par à-coups, selon les envies et les besoins de chacun. Je ne pense pas que ce soit différent des projets logiciels réalisés plus ou moins bénévolement. Il n'y a pas de processus figés lors de la conception, on voit souvent des projets qui ont une méthodologie fixé du type ; review, design, commit ;. Dans le cas de spip, il n'y a rien de tout ça, si quelqu'un à envie de faire quelque chose tout de suite, il se lance, si les autres ne sont pas d'accord, il se fait ; flinguer ;, sinon, il peut discuter sur la liste, c'est au jour le jour. C'est un fonctionnement qui effectivement peut-être difficile à comprendre, certains observateurs, notamment des gens qui arrivent en disant ; je vais prendre le CVS, je vais faire des patches ; ont, au début, un peu de mal à comprendre comment ça marche, ; est-ce que les patches vont être acceptés automatiquement, est-ce qu'il faut que j'en parle avant ;... il n'y a pas de règles fixées, c'est plus une question de ; feeling ;.

Malgré ce coté chaotique et très peu formalisé (voir pas du tout...), on peut quand même dresser des constats, des **tendances émergentes**. On a une version majeure tous les six à douze mois. Alors, attention, par tendances émergentes, je ne dis pas que c'est quelque chose qui surgit automatiquement du fait qu'il n'y a pas de règles, c'est que psychologiquement, même s'il n'y a pas de règles au niveau des sorties de versions, au bout de six mois, si une version n'est pas sortie, on commence à se dire (ou les utilisateurs nous le disent) que ce serait bien d'en sortir une. Il y a quand même une pression qui se fait sentir en interne. Les bugs sont corrigés rapidement, au moins dans la version CVS, on conseil alors aux utilisateurs d'utiliser la version CVS. Ceci d'autant plus que les

versions de développements sont utilisables voir stables car le site du Monde Diplomatique sert de bêta-testeur officiel. Ce site est souvent mis à jour par rapport à la version CVS, et si « le Diplo. » est en version CVS, un petit site peut se le permettre.

Au niveau des commits, il y a beaucoup d'ajouts évidents ou de corrections de bugs, qui se font directement sans qu'il y ait de discussions sur la liste, car on sait qu'ils feront l'unanimité. Cela fait parti de la façon de travailler ensemble, au bout de deux ans, on connaît les gens et on sait comment ils vont réagir à telle ou telle problématique. Mais il y a des cas de grosses fonctionnalités ou de sujets à controverses qui peuvent poser problème, au sens où ils nécessitent discussions et négociations.

La philosophie Spip est un des sujets de discussions et de négociations. On peut le résumer en quelques points qui ne sont pas exhaustifs. Tout d'abord, une cohérence avec les fonctionnalités existantes. Spip, c'est du « full intégré », ce n'est pas pleins de petites briques qui s'assemblent, c'est quelque chose qui est cohérent et qui doit présenter un visage unique à l'utilisateur. Une interface simple et intuitive par souci d'utilisabilité. Eviter l'explosion combinatoire de complexité qui est le syndrome d'outils comme php-nuke, avec des modules dans tous les sens qui permettent de faire tout et n'importe quoi et qui en contre-partie rendent l'utilisation difficile, avec des fonctions trop spécialisées qui servent à deux-trois personnes. Une implémentation correcte et transparente dans le sens où la technique ne doit pas se rappeler au bon souvenir de l'utilisateur avec par exemple des valeurs de messages d'erreurs qui apparaissent parce qu'on a mis une virgule là où ce n'était pas autorisé. Et le souci de compatibilité qui provient du souci de diffusion sur n'importe qu'elle installation, compatibilité avec des installations que l'on peut trouver sur un grand nombre d'hébergements mutualisés type « PHP3 », mesure de sécurité comme la limitation en écriture de fichiers etc...

Le médium officiel du développement est **la liste des développeurs** qui est plus une liste des développements puisqu'il y a trois développeurs officiels et qu'il y a quand même 300 inscrits sur la liste. Elle génère 10 à 20 messages par jour.

Les inscrits sont les trois développeurs, des contributeurs plus ou moins occasionnels qui envoient des patches de correction. On a également un public important de gens qui veulent modifier SPIP dans leur coin soit parce qu'ils sont amateurs de bidouilles soit parce qu'ils ont des clients au sein de leur activité professionnelle qui veulent des fonctionnalités particulières ; et donc, cela leur sert pour connaître mieux le fonctionnement de SPIP, mais aussi, pour savoir par exemple si un type de fonctionnalité est bien dans l'esprit, si cela peut avoir des effets de bord etc... Il y a ceux qui soutiennent SPIP dans le rôle d'utilisateurs avancés qui apportent des informations plus précises, qui émettent des propositions pour les développements futurs, qui donnent leur avis sur la version de développement, qui installent facilement la version CVS. Je pense qu'il s'agit là du profil dominant parmi les gens qui se manifestent. Il y a parmi eux un nombre non négligeable de non-programmeurs, des gens qui ne comprennent rien au PHP mais qui testent quand même les versions de développement.

La liste sert principalement à **régler et à discuter des désaccords** sur les développements. Dès qu'une fonctionnalité ne fait pas l'unanimité parmi les développeurs ou éventuellement les contributeurs les plus réguliers, cette liste permet de régler les désaccords soit par une mise en valeur de la supériorité d'une solution, par exemple par la sécurité d'une solution ou par le fait qu'une des solutions est plus aboutie et répond mieux à la philosophie de SPIP. Cela peut aboutir à un compromis entre plusieurs solutions. Mais il y a beaucoup de sujets qui sont mis en suspend par

absence de consensus ou de relatives convergences de point de vu, car on sait que les solutions proposées ne satisfont pas toutes les exigences. On pourrait dire que c'est l'un des travers de ce mode de fonctionnement. Des échanges parfois âpres, mais n'exagérons rien, ça ne dégénère pas en « flamewar », mais des échanges un peu « café du commerce » sur le côté philosophie SPIP et sur le côté nouvelles fonctionnalités en marge de tout ça ; il y a donc des discussions assez enlevées.

Les contributions au code. Je parle des contributions qui ne proviennent pas des développeurs principaux, donc des gens qui n'ont pas accès au CVS en écriture. Elles sont assez rares. On peut tenter de l'expliquer tout d'abord par les exigences liées à SPIP. Comprendre l'architecture du programme, 30000 lignes de code, ce n'est pas forcément énorme, mais cela commence à devenir respectable ; ce n'est pas forcément bien architecturé, il y a des endroits où cela devient un peu fouillis.

Ensuite, très important, c'est d'**avoir assimilé la philosophie SPIP** ; si on propose des patches qui rajoutent des modules à la php-nuke, ils vont partir à la poubelle. Il y a également l'image, PHP a souvent une mauvaise image auprès des bons programmeurs, je ne sais pas si j'ai raison ou pas sur ce point ; il y a beaucoup de gens qui font du PHP comme un langage joujou mais trois heures par mois pour s'amuser ou font cela en attendant de devenir expert d'un autre langage auquel cas ils abandonneront PHP. PHP n'est pas valorisant pour le programmeur de base, au sens geek ou linuxien. PHP à l'image d'un langage pour débutant, comme c'est un langage qui est facile à apprendre ou à assimiler, forcément, c'est un langage pour débutant, ce qui personnellement me semble totalement faux. Il y a également le rechignement de certains informaticiens aux contraintes extra-techniques fortes comme la facilité d'utilisation par exemple, qui sont des contraintes qui ne sont pas naturelles dans la communauté logiciels libres même si il y a des projets comme Gnome qui savent faire avec ce genre de contraintes et qui continuent quand même à avancer.

Il y a **deux types de contributions**. Les patches qui sont majoritairement souvent mal écrits. C'est vrai que nous avons un petit côté, non pas « intégristes », mais on aime bien conserver une certaine cohérence au niveau du code. Quand aux fonctionnalités entières qui ont été proposées, il y en a très très peu, une ou deux, et elles ont été rejetées car elles ne s'intégraient pas à l'outil actuel, on sentait que c'était une pièce rapportée, il n'y avait pas de cohérence avec le reste du programme. On peut remarquer au niveau de nombreuses contributions, les dégâts de PHP-Nuke au niveau du type de programmation des gens, avec des modules. Plein de gens nous disent, il faut faire « des modules, des modules, des modules », alors que SPIP, au contraire, c'est de l'intégration, on a un outil qui est tout en un, ou tout est relié et tout est cohérent. Donc les modules, c'est vraiment à l'opposé de la philosophie. On reçoit souvent des choses où assez peu d'attention a été accordée à l'interface, avec des interfaces un peu bizarres voir inexistantes.

Conclusion

SPIP est un projet atypique dans le paysage du libre, que ce soit sur les priorités et les méthodes de développement, la façon dont fonctionnent les développements, la façon dont sont décidées les fonctionnalités et les ajouts au code, l'aspect philosophie très forte du logiciel (au sens l'esprit du logiciel, ce qui constitue sa signature). Une personnalité marquée, autant du point de vue de l'outil lui-même que du point de vue de la communauté. Un succès certain, à la fois quantitatif (un nombre de téléchargements respectable...) et qualitatif au sens où les gens n'utilisent par SPIP par défaut, beaucoup de gens sont assez enthousiastes à propos du logiciel ; il y a un attachement quasi-affectif, qui ressemble à mon sens, bien que je ne sois pas vraiment qualifié pour en parler, à

ce qui se passe vis-à-vis de Macintosh chez certaines personnes. Ce n'est pas mon ami SPIP, mais presque.

Cependant, on peut dire que l'avenir de SPIP est en question, non pas que l'on veuille arrêter tout de suite, rassurez-vous, mais au sens où il est difficile d'attirer les contributeurs qui sachent s'intégrer à l'esprit de « la chose ». L'originalité de SPIP est un frein sociologique à l'attraction des contributeurs traditionnels du libre. La question que l'on peut se poser, c'est qui prendra la relève, le jour où deux des trois développeurs originaux auront décidé de faire autre chose.