

Compte rendu du Forum AFUP PHP 2006

par [Julien Pauli](#)

Date de publication :

Dernière mise à jour :

Jeudi et vendredi 9 et 10 Novembre, s'est tenu le forum AFUP 2006. Comme les années précédentes, ce fut un évènement rondement mené, avec beaucoup d'excellentes conférences et de rencontres. Propos de Damien Seguy, Henry Cesbron Lavau, recueil par Julien Pauli.

- I - Présentation
- II - Programme des journées
- III - Retour de quelques conférences
 - Sécurité des applications PHP
 - Design Patterns et PHP
 - EZ Systems : PHP inside
 - Optimisation / industrialisation d'une plateforme de services basée sur PHP
- IV - Conclusion

I - Présentation

Côté contenu, Rasmus Lerdorf et Zeev Suraski étaient présents. Coté francophone, il y avait des retours d'expérience impressionnants dans l'administration belge, dans les décodeurs Canal Plus, ou encore avec une application de mesure de la qualité logicielle PHPMeter.

Le forum AFUP proposait de même un petit salon d'exposants qui regroupait ez Publish, Zend Technologies, Waterproof et isens. Cela a été un succès, avec un niveau élevé d'activité entre deux conférences. De plus, la salle était bondée durant les deux jours. À mon sens, ce n'est qu'un cri de plus en faveur d'une conférence plus grande l'an prochain. L'AFUP doit pouvoir accueillir sans problème 250 personnes sur deux jours, et peut-être même se payer le luxe de deux sessions en parallèle. La liste des grandes entreprises et les institutions gouvernementales qui étaient présentes prouve que PHP intéresse beaucoup de monde !

Le forum AFUP est aussi une plateforme de distribution de cadeaux : un t-shirt, une chemise ainsi qu'un calepin avec son stylo, sans compter quelques vieilles goodies des années précédentes, ainsi que des livres et un memento.

Une autre activité sympathique de l'AFUP est la traditionnelle photo souvenir : cette année, nous avons rassemblé la totalité des visiteurs sur le parvis de la Société Nationale d'Horticulture de France. Il y a deux ans nous étions sur trois rangs, cette année nous étions sur six rangs. Le photographe avait presque du mal à tout cadrer#

II - Programme des journées

- **9h - 9h15** : Keynote
 - **9h20 - 10h20** : PHP en 2006 : Writing and profiling rich applications
 - **10h30 - 11h30** : Ajax et Web Services en PHP : Google AdWords API avec APillity
 - **11h40 - 12h40** : PHP dans l'entreprise... La contribution de ZEND
 - ----- Pause déjeuner -----
 - **14h - 14h45** : PHP au Service Public Fédéral Finances de Belgique : L'intégration réussie PHP-Java
 - **14h55 - 15h40** : Optimisation / industrialisation d'une plateforme de services basée sur PHP
 - **15h50 - 16h30** : Des briques Open Source pour refondre un site internet en PHP (Trac, SVN, Symfony, architecture) -- L'exemple de Richelieu Finance
 - **16h40 - 17h25** : eZ systems : PHP inside
-
- **9h - 9h15** : Keynote
 - **9h20 - 10h20** : eZ components, RAD pour PHP
 - **10h30 - 11h30** : Design Patterns & PHP
 - **11h40 - 12h40** : PHPUnit : Comment faire du développement agile, un atout compétitif
 - ----- Pause déjeuner -----
 - **14h - 15h00** : Sécurité des applications PHP
 - **15h10 - 16h10** : PHPMeter, un outil simple et léger pour évaluer la qualité d'un code PHP
 - **16h20 - 17h20** : Unicode : une révolution en marche pour PHP6

III - Retour de quelques conférences

Sécurité des applications PHP

Le développement a pour but premier le respect des fonctionnalités demandées. Mais il est tout aussi important qu'une application ne fasse que ce qui est attendu. Or il arrive que, notamment sous l'effet d'attaques externes, les applications permettent de faire autre chose que ce pourquoi elles ont été conçues.

C'est là qu'interviennent les indispensables mesures de sécurité.

Damien Séguy, éditeur de nexen.net et conseiller pour nexenservices, attire notre attention sur les failles de développement pouvant amener l'exploitation induite des ressources, la destruction ou le détournement des données ou du site lui-même.

La sécurité commence par une bonne configuration : mise à off de `register_globals` (qui de toutes façons sera supprimé de PHP6, tout comme `magic_quotes` et `safe_mode`), utilisation d'`open_basedir`, `enable_dl` à off, etc.

Et dans l'application même, il faut bien se rappeler qu'**`eval()` is `evil()`**.

Il faut aussi parer aux injections possibles de SQL : par exemple si le champ attend une adresse email, il faut vérifier que l'expression de la saisie est une adresse bien formée. Cela évite que du code SQL n'y soit inclus.

Le Cross Site Scripting (XSS) est une ouverture dangereuse dans la mesure où l'on peut détourner les scripts ainsi appelés.

Quant aux attaques de Cross Site Request Forgery (CSRF) qui exploitent la vulnérabilité d'un site pour permettra à un utilisateur de ce site d'attaquer un autre site, il faut s'en protéger par un haut niveau d'identification.

On aura compris que c'est l'ouverture même du Web qui fait qu'aucune solution n'est définitive.

Et sur toutes ces attaques, ainsi que bien d'autres (email, injection de HTTP, LDAP, ABAP ou AJAXI), les conseils que Damien Seguy nous donne s'avèrent très précieux.

Téléchargez [le diaporama](#).

Visionnez [la vidéo](#).

Design Patterns et PHP

Gérard Croës (de Copix) nous a présenté un outil méthodologique important : les Design Patterns en PHP.

Lancés initialement par l'architecte Christophe Alexander en 1977, les Design Patterns ou [modèles de conception](#) sont apparus en informatique avec la production du GoF (gang of four) qui publie en 1994 "Elements of reusable Object-Oriented Software".

Il s'agit de fournir des solutions à des problèmes classiques de conception.

Dejà, le simple fait de nommer ces solutions permet une communication plus concise entre développeurs : dire *Tu feras une factory qui appellera un singleton*, est plus concis, mais en même temps plus précis que *Tu feras une*

classe qui mettra en oeuvre les paramètres adaptés à la plateforme et qu'il ne faudra appeler qu'une seule fois lors de l'exécution.

Chaque pattern porte donc un nom, décrit le problème concerné et sa solution, cite des exemples d'applications et indique ses limites d'utilisation.

Les Patterns sont habituellement classés en trois catégories : Créateurs, Structuraux et Comportementaux.

Gérald nous a ensuite énuméré les 23 patterns GoF en faisant ressortir les plus utilisés dans sa pratique : le Factory et le Singleton pour les Créateurs, la Façade pour les Structuraux, et en ce qui concerne les Comportementaux, l'Observer et le State.

On peut enrichir la liste des patterns mais, après avoir rencontré et résolu de la même manière au moins 3 fois un problème particulier pour pouvoir l'élever valablement au rang de Pattern.

Il ne faut cependant pas faire du Design Pattern à tout crin.

Les Anti-patterns donnent, *a contrario*, des cas d'actions à ne pas faire.

L'ensemble fournit un cadre de bonnes pratiques dans la structuration des applications.

Téléchargez [le diaporama](#).

Visionnez [la vidéo](#).

EZ Systems : PHP inside

Pour terminer cette première journée fonctionnelle, Roland Benedetti, chargé de l'Europe Sud nous a présenté l'offre d' "eZ systems". Le produit phare de cette société est incontestablement le CMS haut de gamme "eZ publish".

Le produit continue d'exister aussi en version entreprise avec engagement total de l'éditeur. "eZ publish" est basé sur les "eZ components" présentés le jour suivant.

La société propose aussi des solutions plus spécifiques basées sur ses outils, ainsi que la formation, du support et de l'expertise.

Sur les 6 années de son développement, le CMS "eZ publish" a été téléchargé 2 000 000 de fois. On le trouve au cœur de solutions corporate, d'e-commerce, de portails, communautaires, documentaires ou d'internets / extranets.

Par rapport à ses concurrents, il permet en particulier la gestion du versionning, les publications cross-media et multi-sites. Il possède par ailleurs toutes les propriétés des grands CMS. Très ouvert en terme d'interopérabilité, très flexible, intégrant naturellement Open Office, il est bien architecturé par la séparation contenu / présentation.

La structure des contenus ne se limite pas aux articles, mais peut prendre en compte toute une hiérarchie de zones spécialisées d'un document avec de surcroît l'héritage des structures comme en POO.

Pour les développeurs qui veulent l'utiliser comme framework, PHPEdit de WaterProof est l'éditeur qui fournit un environnement complet avec debugger.

Pour terminer, Roland nous a présenté le site de Rhône Tourisme développé en un mois, ainsi qu'un projet plus important, celui d'Edipresse. Ce dernier a pu être rebâti en 14 mois (au lieu des 18 mois prévus initialement). Une démarche industrielle telle qu' "eZ publish" le permet était requise pour le plus gros site de presse en ligne suisse.

Ces deux projets sont révélateurs de l'étendue de la gamme de solutions couverte par le CMS "eZ publish".

Téléchargez [le diaporama](#).

Consultez le [test de PHPEdit](#), par Guillaume Rossolini.

Optimisation / industrialisation d'une plateforme de services basée sur PHP

Le succès du PHP porte à l'employer dans des solutions de plus en plus diverses et critiques. C'est pourquoi il est nécessaire de mettre en oeuvre de bonnes pratiques d'optimisation au niveau applicatif. Cyril Pierre de GEYER, responsable technique d'Anaska et co-auteur du livre "PHP5 avancé", a commencé à présenter la solution " audit et performances " d'Anaska qui est une prestation d'audit de plateformes qui permet de pointer du doigt toutes les améliorations possibles.

- Qualité du code (structure / objet) ;
- Appels aux bases de données (optimisation des connexions, plan de requêtes : EXPLAIN en MySQL) ;
- Phases : environnement de développement, pré-production, production ;
- Performances (nombre de requêtes / seconde, stabilité) ;
- Sécurité ;
- Maintenance.

Guillaume Ponçon, également de la société Anaska et auteur du livre "Best practices PHP 5", a enchaîné en présentant les outils permettant de mener à bien ce type d'audit : XDebug + KCacheGrind et la "Zend platform". Ces outils permettent de repérer les opérations coûteuses et les doublons qui conduiront à factoriser le code.

- Architecture (répartition de charges) ;
- MySQL (réplication) ;
- Cache d'OpCode ;
- Cache de contenu (Jpcache) ;
- Compression de contenu (mod_gzip).

La dernière partie, présentée par Franck Lombardi, responsable technique TV ADSL de Canal Plus, a permis de voir une application critique et non Web utilisant PHP. Saviez-vous qu'en zappant de chaîne sur votre TVADSL vous faites des requêtes PHP ? Saviez-vous qu'en consommant de la TV à la demande vous utilisez PHP ? Très probablement non et c'est tout le public qui a été intéressé par cette conférence peu commune. Franck nous a détaillé comment PHP avait été choisi au début pour répondre vite à un besoin et comment, au fil et à mesure du temps, l'application avait évolué vers une industrialisation grâce notamment aux conseils d'Anaska.

Téléchargez [le diaporama](#).

IV - Conclusion

Bravo au bureau de l'AFUP pour la réalisation du sixième forum PHP. Les conférences étaient de bon niveau, mais aussi surprenantes et originales ; l'ambiance était très détendue et propice à un réseautage important entre les utilisateurs PHP, les conférenciers et les entreprises de services présentes.

- [Toutes les vidéos](#) ;
- [Les photos](#);
- [Les autres résumés](#).