

# Le dispositif ezPAARSE/AnalogIST : pour une analyse mutualisée des logs d'accès aux ressources documentaires payantes

## Thomas Porquet

Département études et prospective - Consortium Couperin  
Service Commun de la Documentation de l'Université Paris Descartes  
45 rue des Saints Pères  
75006 PARIS

## Dominique Lechaudel

Service Ingénierie de Projets - Pôle Systèmes d'Information  
Institut Information Scientifique et Technique du CNRS  
2 allée du Parc de Brabois  
54519 Vandœuvre-lès-Nancy

## Stéphane Gully

Chargé de Projets - Pôle Systèmes d'Information  
Institut Information Scientifique et Technique du CNRS  
2 allée du Parc de Brabois  
54519 Vandœuvre-lès-Nancy

## Thomas Jouneau

Bibliothèque Numérique - Direction de la Documentation et de l'Édition  
Université de Lorraine  
Île du Saulcy  
57045 Metz Cedex 01

## Résumé

La présentation “Quantifier les accès à la documentation électronique payante : un panorama des enjeux” expose le fort besoin des services communs de documentation de l'ESR pour évaluer l'usage des ressources numériques.

Le duo AnalogIST / ezPAARSE permet de réaliser cette évaluation en exploitant une des caractéristiques techniques de la mise à disposition des ressources numériques : le contrôle d'accès via des proxies.

AnalogIST<sup>1</sup> a pour objectif de décrire et coordonner le travail d'analyse des fichiers de logs générés au sein des établissements par les proxies. Le progiciel libre ezPAARSE (développé à l'INIST-CNRS et écrit entièrement en javascript(1)) y est d'ores et déjà en ligne et fonctionnel.

Le point de départ de ce travail est la présence d'informations précises et exploitables dans ces fichiers de logs bruts, telles que :

- le login, pour l'identification de l'institution et la discipline scientifique de l'utilisateur,
- l'URL du PDF de l'article téléchargé par le login en question, pour l'identification de la revue consultée.

---

<sup>1</sup> <http://analogist.couperin.org>

Le travail d'ezPAARSE (2) consiste à extraire ces informations et les enrichir au sein d'événements de consultation puis de les délivrer sous forme d'un fichier "propre" au format texte CSV qui servira de base à la réalisation de statistiques d'usages locales et maîtrisées au niveau de détail voulu.

Nous souhaitons mutualiser le travail d'analyse des logs : les utilisateurs peuvent déployer ezPAARSE au sein de leur établissement et réaliser leurs propres analyses. Les programmes, appelés "parseurs", chargés de découper les URL imposent une maintenance régulière qui nécessite des compétences documentalistes (analyse des plateformes éditeurs) et informatiques (implantation des programmes).

## Mots-clefs

*Ressources numériques, statistiques d'usage, COUNTER, reverse proxy, ezProxy, méthode agile, SCRUM, git, github, NodeJS, javascript, streaming*

## 1 Introduction et contexte

La problématique de l'évaluation de l'utilisation des ressources électroniques payantes est une question stratégique et récurrente qui ne trouve pas actuellement de réponse commune satisfaisante. Les coûts d'abonnements n'ont cessé d'augmenter, d'autant plus rapidement que les ressources étaient mises en ligne plutôt que diffusées au format papier et le contexte budgétaire est désormais délicat (voir par exemple les décisions récentes de désabonnements prises par les universités de Aix-Marseille 1, Rennes 1, Paris 4 Sorbonne).

Pour ce travail d'évaluation, nous pouvons en partie nous appuyer sur les rapports statistiques quand ils sont fournis par les éditeurs commerciaux, comme c'est le cas avec le projet MESURE<sup>2</sup>, mais ces rapports souffrent de limitations qui en restreignent l'intérêt :

- ils sont difficiles à comparer quand ils ne respectent pas le guide de bonnes pratiques COUNTER<sup>3</sup> qui promeut la standardisation des rapports, leur périodicité et leur mode de diffusion ;
- ils sont difficiles à vérifier : nous n'avons jamais accès à la "matière première" qui sert à constituer les rapports et ne pouvons donc mener aucune étude contradictoire ;
- les données fournies sont en général agrégées par entité facturée (établissement) et ne sont donc pas suffisamment détaillées pour disposer d'informations précises sur les typologies fines de publics qui consultent les ressources (enseignants, chercheurs, étudiants, discipline, etc...).

Pour surmonter les problèmes susmentionnés, il faut se tourner vers une solution de production locale de données statistiques. La solution technique adoptée par l'INIST-CNRS<sup>4</sup> consiste à utiliser huit serveurs mandataires distincts (proxies) pour contrôler les accès et relayer l'information entre les communautés d'utilisateurs et les sites des éditeurs. Elle permet aussi de quantifier les accès réussis aux ressources via l'analyse des journaux de transaction (fichiers de log). Dès 2006 l'INIST met en place un groupe de travail pour développer des outils et des procédures d'analyse de ces fichiers et répondre au besoin récurrent d'évaluer les usages et ainsi justifier les crédits engagés dans la politique documentaire.

Ces enjeux dépassent naturellement le cadre du CNRS et concernent aussi d'autres organismes aux besoins identiques. Ces organismes ont souhaité rejoindre, via un partenariat, les travaux déjà engagés, ce qui a permis d'envisager de les généraliser.

C'est dans ce contexte que le projet de plate-forme nationale AnalogIST/ezPAARSE est lancé en novembre 2012 et renouvelé en septembre 2013, concrétisant un partenariat établi entre Couperin, l'INIST-CNRS et l'Université de Lorraine, avec l'objectif de mutualiser une chaîne de traitements et des procédures déjà opérationnelles à l'INIST.

---

<sup>2</sup> Le projet MESURE est disponible depuis octobre 2012 sur <http://mesure.couperin.org> et moissonne pour les membres de Couperin leurs rapports auprès de certains éditeurs

<sup>3</sup> Pour plus d'information, voir <http://www.projectcounter.org/>

<sup>4</sup> Opérateur pour les communautés CNRS des ressources mises en ligne par les éditeurs scientifiques (via les BiblioSites thématiques)

Une première version complète a été livrée en septembre 2013. Un deuxième cycle de développement a été lancé sur 2013/2014. Deux ingénieurs informaticiens y travaillent à plein temps.

## 2 Description succincte du projet

### 2.1 Le progiciel ezPAARSE

[ezPAARSE](#)<sup>5</sup> est un progiciel qui permet l'analyse, l'enrichissement et l'exploitation des logs d'accès aux ressources numériques proposées par les plateformes Web des éditeurs de littérature scientifique. Il se présente sous la forme d'une application Web disposant d'un formulaire HTML et d'une API permettant l'injection manuelle et automatique des logs en question.

ezPAARSE est capable d'analyser [différentes plateformes éditeur](#)<sup>6</sup>.

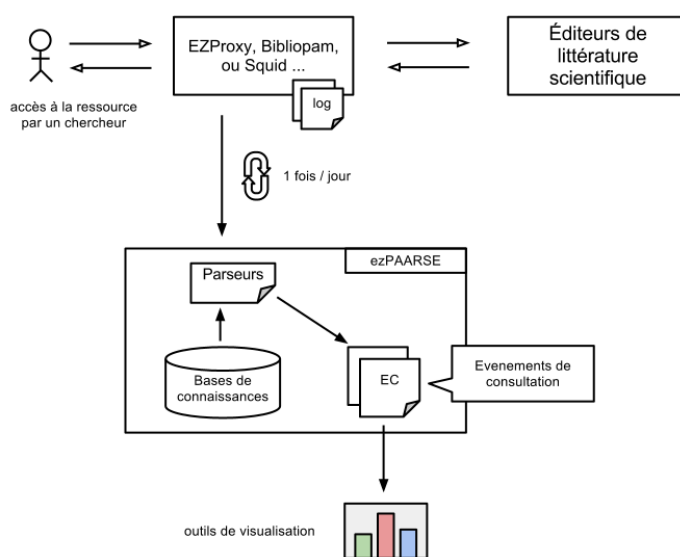


Figure 1 - Schéma simplifié du fonctionnement d'ezPAARSE. Le serveur mandataire (reverse proxy de type EZproxy) agrège l'accès aux ressources numériques des éditeurs et réalise les contrôles d'accès. Il journalise l'ensemble des transactions dans des fichiers de log. Les fichiers de log sont analysés par ezPAARSE à l'aide des parseurs spécifiques à chaque plateforme éditeur pour produire des événements de consultation (EC). Les EC sont stockés sous forme de fichier texte CSV exploitable dans des outils de reporting de type Excel ou Omniscope Visokio.

ezPAARSE sert de brique essentielle dans la création de la plateforme AnalogIST<sup>7</sup>, à périmètre national, synchronisable avec des instances locales, installées au sein des institutions intéressées.

### 2.2 La plateforme nationale AnalogIST

D'ores et déjà mise en ligne<sup>8</sup>, AnalogIST est un lieu de mutualisation où sont publiées :

- la [procédure permettant d'analyser une plate-forme éditeur](#)<sup>9</sup>, condition préalable à l'écriture d'un parseur et réalisables par des documentalistes ;

<sup>5</sup> ez : easy / PAARSE : Progiciel d'Analyse des Accès aux Ressources Electroniques

<sup>6</sup> <http://analogist.couperin.org/platforms/>

<sup>7</sup> AnalogIST est l'acronyme de "Analyse des Logs de l'Information Scientifique et Technique"

<sup>8</sup> sur <http://analogist.couperin.org>

<sup>9</sup> <http://analogist.couperin.org/platforms/contribute/>

- les différentes analyses déjà réalisées par la communauté ESR. Elles restent disponibles et modifiables pour suivre les évolutions des plateformes et permettre la mise à jour des parseurs.

C'est tout un travail de veille et d'analyse qui se trouve ici publié et tenu à jour. Une instance fonctionnelle d'ezPAARSE y est hébergée pour les établissements qui ne peuvent ou ne souhaitent pas en installer une instance locale.

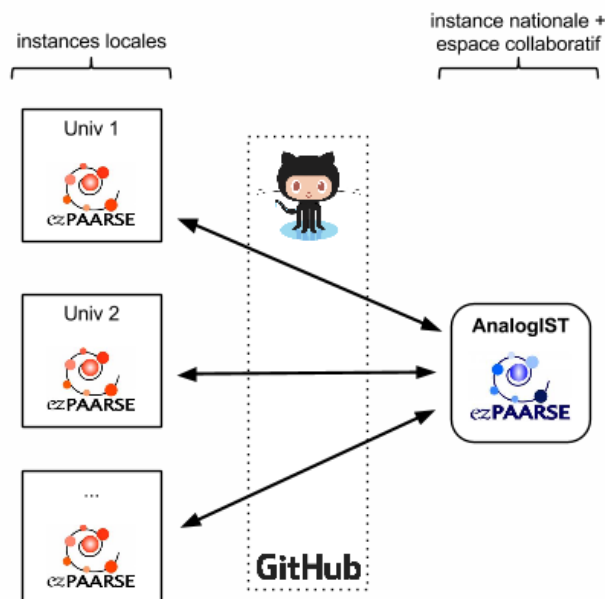


Figure 2 - Gestion des interactions entre différentes instances ezPAARSE. Les instances locales peuvent faire leur mise à jour de parseurs ou de base de connaissance et se synchroniser via un passage par le système de gestion de version (SGV) GitHub

### 3 Méthodologie

L'expérience acquise au cours du développement d'une version précédente de logiciel d'évaluation d'usage des ressources numériques propre à l'INIST-CNRS a servi de base méthodologique pour la mise en œuvre de ce projet, orienté cette fois vers la mutualisation et les partenariats (3).

#### 3.1 Besoin d'évolution rapide

Le besoin de pouvoir présenter rapidement des résultats tangibles mis en œuvre par une équipe mixte (développeurs, informaticiens système, documentalistes) nous a fait choisir une méthode de développement agile, SCRUM<sup>10</sup> (4). Cette méthode est caractérisée par un fonctionnement sous forme de cycles de développements courts avec livraisons régulières de versions fonctionnelles du logiciel, et ce dès la première itération.

Pour initialiser le projet il est nécessaire de produire un document court chargé de définir l'objectif commun à tous les partenaires du projet. Ce document est appelé : la vision du produit. Dans le cas d'ezPAARSE, la vision du produit a été rédigée au démarrage du projet après un remue-méninges de l'ensemble des partenaires du 2 au 5 mai 2012. Cette réunion a également permis de produire un autre document : le "product backlog" qui contient la liste des fonctionnalités à développer. Le product backlog est une liste priorisée qui évolue au cours du temps de façon à tenir compte des retours utilisateurs et des nouvelles idées au cours du projet.

La livraison rapide de versions fonctionnelles permet d'avoir des retours utilisateurs immédiats et d'orienter sans cesse le projet vers les "vrais" besoins des utilisateurs. On peut faire le parallèle entre les résultats produits par cette méthode et une série télévisée, dont les épisodes se succèdent régulièrement et sont regroupés en saisons. L'intérêt des

<sup>10</sup> [http://fr.wikipedia.org/wiki/Scrum\\_\(m%C3%A9thode\)](http://fr.wikipedia.org/wiki/Scrum_(m%C3%A9thode))

utilisateurs est maintenu tout au long du projet, leur implication grandit au fil des développements et ils peuvent influencer sur le scénario. Dans le cas d'ezPAARSE, la saison 1 à l'issue de laquelle a été livrée la version 1.0 s'est déroulée sur une année en 10 itérations correspondant chacune à une version du logiciel, chaque version ayant une thématique. Le fil rouge est la reconnaissance progressive d'un nombre croissant de plateformes d'éditeurs. La saison 2 est en cours et durera, elle aussi, une année.

## 3.2 Besoin de collaboration et de contributions

Le souci de faciliter la collaboration autour du code source se manifeste dans la structure même d'ezPAARSE, conçue pour pouvoir intégrer la reconnaissance de nouvelles plateformes à la manière de greffons. Le cœur d'ezPAARSE (voir Figure 1 - ) met en œuvre une succession de parseurs qui sont de petits programmes capables d'analyser la structure des URL spécifiques à une plateforme éditeur. Chaque parseur est capable de fonctionner indépendamment, hors du contexte ezPAARSE, et peut donc être développé par un contributeur extérieur. Le langage de développement n'est pas imposé, même si la plupart des parseurs sont écrits en Javascript.

Dans cette optique, il était logique de rendre le code source disponible à la communauté sous forme de logiciel libre et de le protéger par la licence de logiciel libre CeCILL préconisée par le CEA, le CNRS et l'Inria. La caractéristique contaminante de cette licence confère une garantie suffisante et permet une appropriation du code par les contributeurs.

## 3.3 Besoin d'interaction à distance

Il était nécessaire de disposer d'un Système de Gestion de Version (SGV) à distance pour le code source. Les systèmes de gestion de version de type Git (5) et leur interface comme GitHub (6) permettent la collaboration d'un ensemble de développeurs. C'est le cas pour ezPAARSE : la majorité des développements se font à l'INIST-CNRS à Nancy, les partenaires de l'Université de Lorraine et de COUPERIN interviennent ainsi que les contributeurs (pour les parseurs) des Universités de Rennes et Saint-Etienne.

L'architecture du progiciel ezPAARSE prévoit le déploiement d'instances locales qui pourront fonctionner en interaction avec une instance nationale (voir Figure 2 - ). L'usage du SGV a été étendu pour permettre la gestion de cette synchronisation. Des dépôts spécifiques sont réservés aux parseurs et aux bases de connaissances. La synchronisation se fait par l'intermédiaire d'une interface d'administration propre à chaque instance et régulée via le SGV.

# 4 Implémentation technique

## 4.1 Chaîne de traitement

Pour commencer, ezPAARSE attend en entrée des lignes de logs issues d'un proxy ou reverse-proxy. Ces lignes de logs sont découpées à la volée (en streaming) pour permettre le traitement de grandes quantités sans risquer de saturer la mémoire. Plusieurs formats de logs sont gérés par défaut (pour différents types de proxies). Un paramétrage avancé du format est également possible. Chaque élément de la ligne de log est isolé pour servir dans la poursuite de l'analyse.

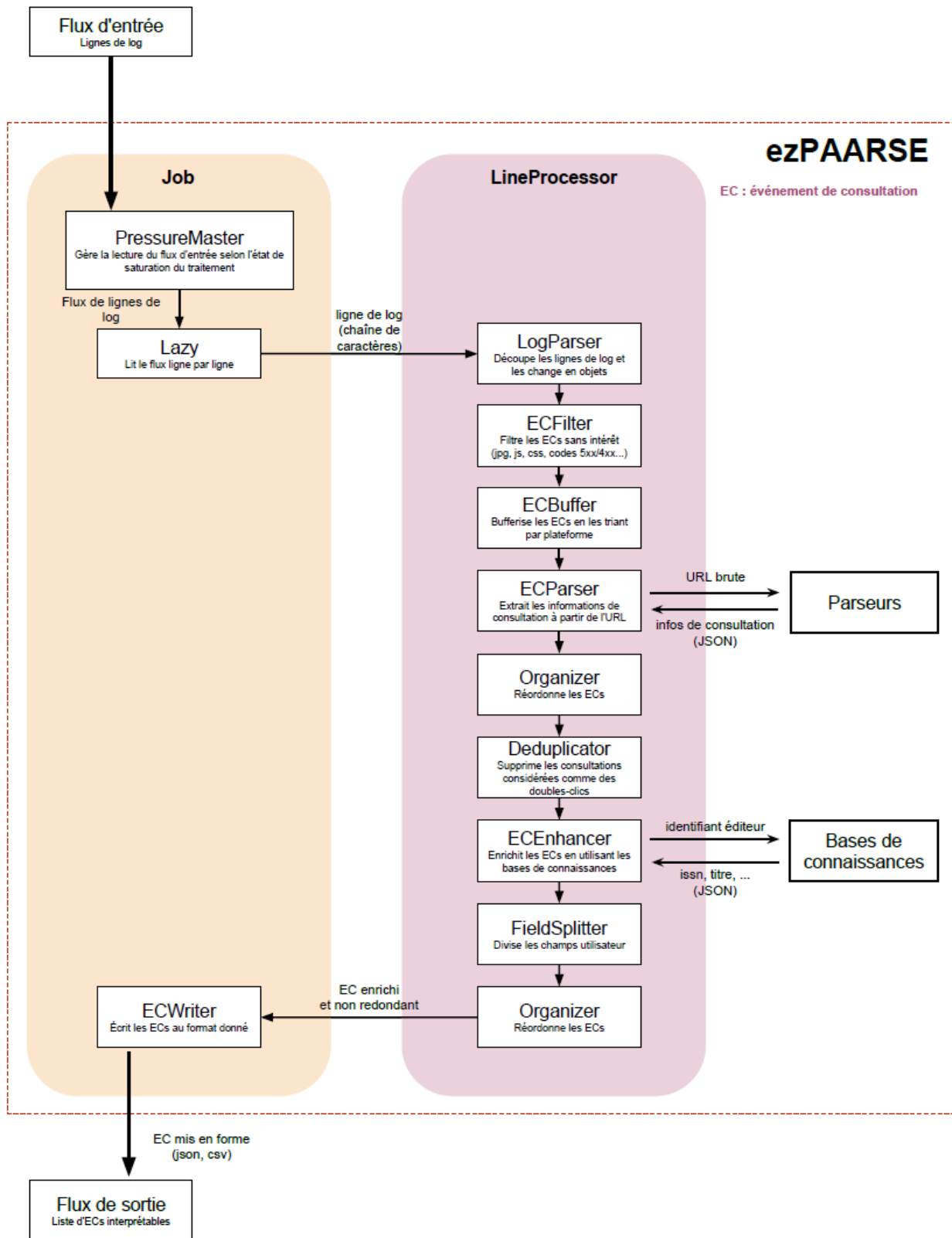


Figure 3 - Fonctionnement interne d'ezPAARSE. Le flux d'entrée est constitué d'un flux de lignes de log. Il est pris en charge par un job régulant via des évènements la saturation de ce flux en fonction de la vitesse du traitement (« back pressure »). Les lignes sont découpées, la plateforme d'origine est reconnue de façon à utiliser le parseur adéquat. Les consultations considérées comme doublons sont éliminées et l'EC est enrichi à l'aide des bases de connaissance. Le flux de sortie est constitué d'EC au format CSV ou JSON.

L'étape suivante consiste à découper le champ URL des lignes pertinentes<sup>11</sup>. ezPAARSE classe ces URL dans des groupes. Chaque groupe correspond à une plate-forme d'éditeur et doit être analysé par un parseur adapté, chargé d'extraire des informations d'identification<sup>12</sup> en utilisant des expressions régulières (7). En effet, la sémantique des URL, reflet des choix techniques des éditeurs ou de leurs sous-traitants, est différente d'une plate-forme à l'autre.

Pour chaque ligne de log, ezPAARSE crée un "événement de consultation" et l'enrichit si besoin. Par exemple, si l'ISSN de la revue consultée n'est pas disponible dans l'URL traitée, ezPAARSE fait appel à des "bases de connaissances éditeurs"<sup>13</sup> qui établissent la correspondance entre un identifiant propre à l'éditeur et son ISSN.

Puis ezPAARSE procède à l'enrichissement relatif aux données administratives. Le login ou l'adresse IP sont des informations qui peuvent, une fois croisées avec un annuaire LDAP institutionnel (ou autre référentiel utilisateur), dévoiler la discipline scientifique, le laboratoire, etc... . L'événement de consultation est alors complet et peut sortir d'ezPAARSE au format CSV ou JSON.

Nous pourrions alors appliquer à ce flux un filtre temporel<sup>14</sup> (8) (9) pour tenter d'éliminer les phénomènes de double-clics (Figure 4 - ).



Figure 4 - Algorithme de dédoublement des événements de consultation (EC) selon COUNTER. (1) Une fenêtre temporelle regroupe plusieurs EC triés chronologiquement (20s) et l'entrée du nouvel EC (en pointillé en

<sup>11</sup> les éléments non-significatifs, comme les appels d'images ou les feuilles de style, sont ignorés

<sup>12</sup> comme par exemple l'ISSN d'une revue

<sup>13</sup> appelées PKB (pour Publisher Knowledge Base)

<sup>14</sup> comme celui que propose le guide de bonnes pratiques COUNTER dans sa section "Return codes and time filters" (voir <http://www.projectcounter.org/r4/COPR4.pdf>, p.25)

haut) fait avancer cette fenêtre. (2) Les EC distincts (uniques) dépassant la fenêtre temporelle (>20s) sont considérés comme valides au sens COUNTER (non double cliqués). (3) Le nouvel EC entré dans la fenêtre temporelle en étape 1 peut alors être dédoublonné avec les autres EC présents dans la fenêtre. (4) Les EC restants sont comparés aux prochains qui arriveront chronologiquement

A ce stade, les événements de consultations sont exploitables et fournissent une base de travail pour la création d'indicateurs.

L'étape ultime de la chaîne de traitement, hors du périmètre d'ezPAARSE, est la visualisation de ces données tabulaires avec un outil avancé de visualisation et de reporting (par exemple : Excel, LibreOffice, Visokio, etc...) (Voir Figure 5 - Figure 6 - )

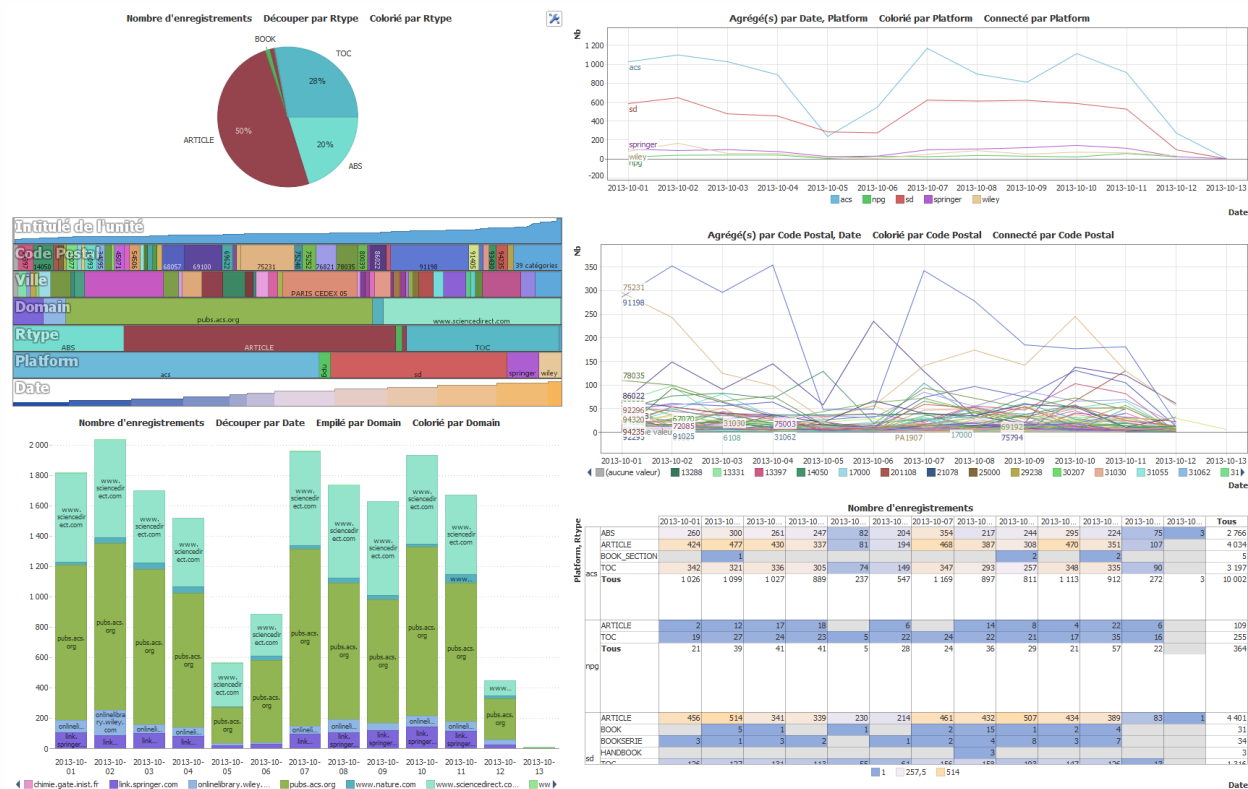


Figure 5 - Exploitation des EC dans des outils de reporting. Visualisation utilisant l'outil Visokio d'un traitement réalisé par ezPAARSE



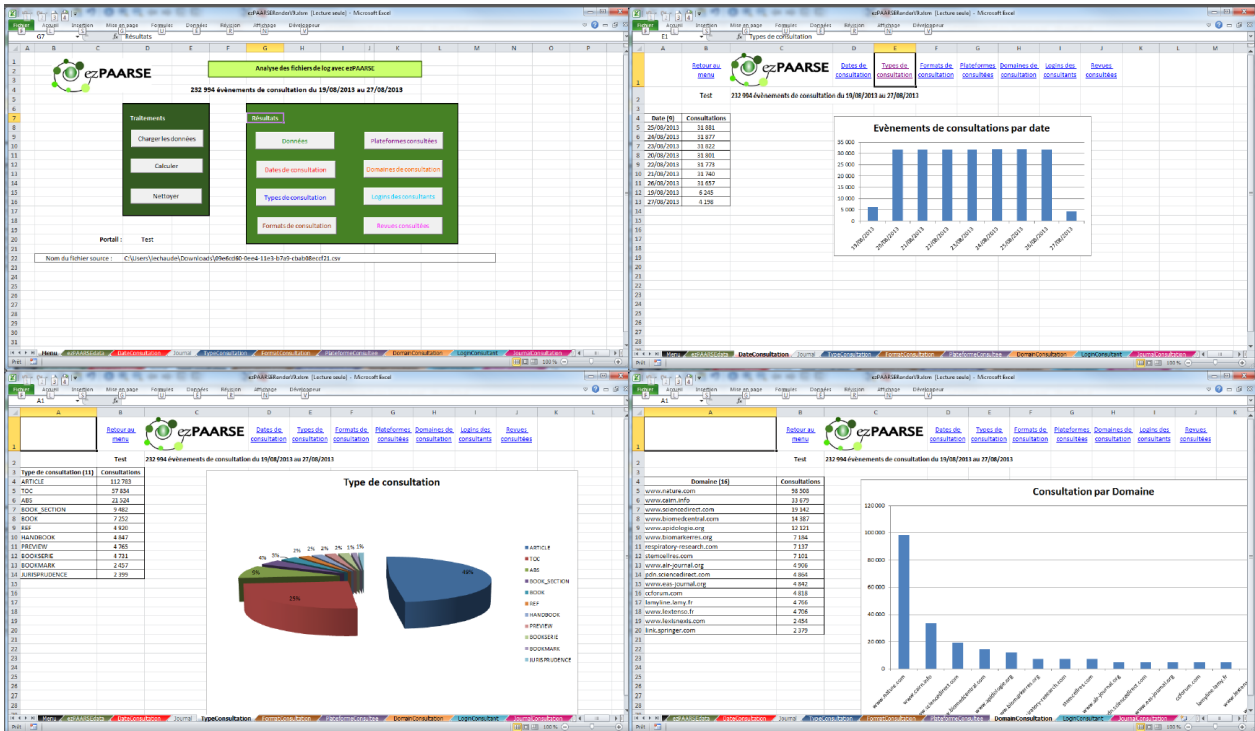


Figure 6 - Exploitation des EC dans des outils de reporting. Visualisation utilisant le fichier Excel de rendu mis à disposition dans la distribution ezPAARSE

## 4.2 Automatisation

ezPAARSE est construit autour d'un Web service **REST**. Il est ainsi possible d'envoyer des données à ezPAARSE en utilisant des outils tels que cURL ou bien à partir des langages de programmation qui disposent d'un client HTTP.

Un cas d'usage qui illustre l'intérêt d'une telle architecture est le suivant :

- Le logiciel logrotate est utilisé pour archiver périodiquement les logs du proxy (EZproxy, Squid, BiblioPAM ...)
- A la fin de son traitement, logrotate lance la commande **curl** avec comme destination le Web service d'ezPAARSE et comme paramètre le fichier de log qui vient d'être archivé.
- Le résultat du curl est ensuite récupéré et peut soit être envoyé par courriel à la personne responsable des statistiques, soit archivé dans un fichier sur le système.

## 4.3 Architecture

Nous avons conçu le progiciel pour faciliter conjointement son déploiement (installation d'instances multiples) et rendre le plus accessible possible le développement des parseurs.

Les parseurs sont des éléments clés pour générer et enrichir les événements de consultation. Leur développement est humainement coûteux mais hautement mutualisable, puisque beaucoup d'établissements sont abonnés aux mêmes plateformes.

Nous les avons conçus sous forme de greffons. Ainsi, ils sont autonomes. Ce sont des commandes qui ingèrent un flux d'URLs et retournent les éléments sémantiques extraits dans un format commun à tous les parseurs. Chaque parseur est

accompagné de tout ce qui est nécessaire à son fonctionnement et à sa validation. Le fonctionnement sous forme de commande enchaînée (“pipeable”) autorise l’utilisation de plusieurs langages de programmation<sup>15</sup>.

## 4.4 Qualité logicielle

Comme le code des parseurs est ouvert aux contributions, nous avons fourni un effort particulier pour intégrer des tests systématiques à chacun des éléments ajoutés.

Au cours des développements, un serveur d’intégration continue vérifie en permanence la validité du déploiement et du fonctionnement du logiciel à travers des tests unitaires. Nous avons utilisé l’outil [Travis CI](#)<sup>16</sup> pour cela. Dès qu’au moins un des tests unitaires échoue, Travis envoie un rapport d’erreur signalant qu’ezPAARSE est cassé dans un message à l’équipe de développement.

Nous disposons d’environ 60 tests unitaires qui sécurisent les fonctionnalités d’ezPAARSE. A chaque ajout d’une fonctionnalité, nous lui associons un ou plusieurs tests ainsi que sa documentation. Cette démarche nous permet de disposer à tout moment d’un logiciel stable livrable fréquemment.

L’auto-documentation du progiciel renforce la facilité d’usage et guide la conduite des développements contributifs. La documentation d’ezPAARSE est générée avec l’outil [beautifuldocs](#)<sup>17</sup> et le site AnalogIST est réalisé avec l’outil [Dokuwiki](#)<sup>18</sup>.

Pour permettre de faciliter la diffusion et l’installation d’ezPAARSE, chaque nouvelle version est livrée dans différents formats pour les différentes plate-formes : zip, tar.gz, deb, rpm, exe.

## 5 Conclusion

La communauté de l’ESR exprime une attente forte et souhaite depuis longtemps disposer d’indicateurs produits de façon homogène et comparables qui serviront lors des négociations et du pilotage des politiques documentaires (10). Nous pensons que le projet AnalogIST/ezPAARSE est un élément de réponse concret à cette demande.

Nous pointons l’importance de conserver un lien étroit entre les services informatiques et les problématiques propres à l’accès aux ressources numériques payantes. Ces problématiques sont consommatrices en temps humain. Les réponses passent par la mutualisation d’éléments clés au cœur de ce projet.

La phase 2 du projet ezPAARSE a débuté en septembre 2013 et se terminera en septembre 2014. Les grandes orientations de cette seconde phase sont les suivantes : prendre en compte un maximum de plateformes d’éditeurs de littérature scientifique ; encourager et faciliter le déploiement d’ezPAARSE dans les centres de documentation de l’ESR ; optimiser les processus de contribution à ezPAARSE et AnalogIST ; et réaliser un prototype illustrant une autre manière de mesurer les usages (technique des trackers).

Lors de la première phase, la pratique de la méthode agile SCRUM s’est révélée être un succès. Cette méthode est logiquement appliquée pour la réalisation de la seconde phase. Des versions stables du progiciel continueront donc d’être régulièrement livrées (environ une par mois). A ce titre, il est nécessaire de rappeler que les retours d’utilisateurs d’ezPAARSE sont très appréciés. Ce sont des éléments précieux qui permettent à l’équipe d’orienter les développements au plus près des besoins. Pour se tenir informé de l’avancée des développements et pour communiquer avec l’équipe, le compte twitter [@ezpaarse](#)<sup>19</sup> ainsi qu’une adresse mail dédiée [ezpaarse@couperin.org](mailto:ezpaarse@couperin.org) sont à disposition.

---

<sup>15</sup> javascript, php, perl, python...

<sup>16</sup> <https://travis-ci.org/ezpaarse-project/ezpaarse>

<sup>17</sup> <http://beautifuldocs.com/>

<sup>18</sup> <https://www.dokuwiki.org>

<sup>19</sup> <https://twitter.com/ezpaarse>

## 6 Bibliographie

1. Node JS Platform [Internet]. Available from: <http://nodejs.org/>
2. 31 ezpaarse-project J on O, Members 2012 6 Public Repos 0 Private Repos 7. ezpaarse-project [Internet]. [cited 2013 Oct 17]. Available from: <https://github.com/ezpaarse-project>
3. Anne-Marie BADOLATO, Magali COLIN, Philippe HOUDRY, Dominique LECHAUDEL, Sonia LAUNAY. RESSOURCES ELECTRONIQUES DES PORTAILS DE L'INIST : ANALYSE QUALITATIVE PAR COMPARAISON DES CONSULTATIONS ET DES FACTEURS D'IMPACT [Internet]. Available from: <http://atlas.irit.fr/PIE/VSSST/Actes-VSSST2009-Nancy/Session-7/Badolato%20Colin%20Houdry%20Launay%20Lechaudel.pdf>
4. Ken Schwaber (virman@aol.com). Scrum Development Process [Internet]. 1995. Available from: <http://www.jeffsutherland.org/oops/schwaber.html>
5. GIT - Documentation [Internet]. Available from: <http://git-scm.com/documentation>
6. GitHub [Internet]. [cited 2013 Oct 17]. Available from: <https://github.com/>
7. Regular Expressions - MDC Docs [Internet]. [cited 2011 May 10]. Available from: [https://developer.mozilla.org/en/JavaScript/Guide/Regular\\_Expressions](https://developer.mozilla.org/en/JavaScript/Guide/Regular_Expressions)
8. COUNTER code de bonnes pratiques pour les ressources électroniques Version 4 [Internet]. COUPERIN - [www.couperin.org](http://www.couperin.org); Available from: [http://couperin.org/images/stories/documents/Statistiques/COUNTER/V4\\_FR/appd\\_fr.pdf](http://couperin.org/images/stories/documents/Statistiques/COUNTER/V4_FR/appd_fr.pdf)
9. The COUNTER Code of Practice for e-Resources: Release 4 [Internet]. COUNTER Online Metrics; 2012. Available from: <http://www.projectcounter.org/r4/COPR4.pdf>
10. Petitjean A, Lamouroux M, Tête C, Audureau L, Devaud C, Falconnet M, et al. Méthodes techniques et outils. Doc-Sci Inf. 2013 Apr 15;Vol. 50(1):8–15.