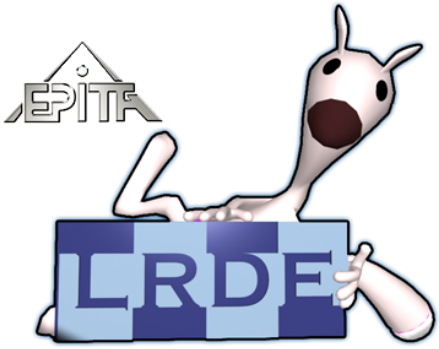


L'air de rien
N° 23

L'aléastriel du **L**aboratoire de **R**echerche et de **D**éveloppement de l'**E**PITA [1] Numéro 23, 10 Octobre 2011

Édito

par *Olivier Ricou (Enseignant-Chercheur)*

Le mois dernier je vous ai presque annoncé l'arrivée d'Etienne et de Benjamin au sein du LRDE. Aujourd'hui voici leurs mini-bios et nul doute qu'au prochain numéro ils vous expliqueront plus en détail leur sujet de recherche. Ce numéro est aussi l'occasion de mettre en valeur un étudiant du LRDE un peu fou, il aime Javascript, mais fort sympathique et qui sait partager sa passion. Il le fait à travers un article ici, mais aussi le mardi 25 octobre à 18h30 dans l'amphi masters (entrée libre). Enfin je ne résiste pas au plaisir de vous proposer encore du Scribo. Ce projet qui nous a tenu en haleine pendant des années et qui a permis à Guillaume de s'épanouir dans le traitement d'image, est fini. Il a permis d'enrichir notre plateforme Olena à tel point qu'on n'a pas eu d'autre choix que de sortir la version 2.0. Avec cette nouvelle version commence notre conquête du Monde. Actuellement Olena est dans la distribution de Linux Mandriva pour ajouter des fonctionnalités au bureau intelligent (ou sémantique) Népomuk de KDE. Nous travaillons pour que d'autres distributions l'intègrent et nous visons aussi d'autres applications. Par exemple un greffon pour Gimp serait probablement une bonne source de contributeurs au projet.

Mini-bios de Benjamin et d'Etienne

Benjamin Raynal

Benjamin est docteur de Paris-Est Marne-la-Vallée. Ses travaux portent principalement sur la captation de mouvement en temps réel à partir d'ensembles de flux vidéos. Cela peut servir notamment à la conception de nouvelles interfaces homme-machine, la vidéo surveillance automatique, ou encore en domotique.

Il intègre le LRDE en tant que Post-doc et s'intéresse actuellement à l'extraction automatique de texte dans les images.

Etienne Renault

Diplômé de Paris VI en système et applications réparties, Etienne s'intéresse à la vérification formelle des systèmes concurrents. Il intègre cette année le LRDE dans le cadre de sa thèse portant sur la composition dynamique de techniques pour le model checking efficace. Cette thèse vise à introduire de nouvelles méthodes permettant de s'attaquer au problème de l'explosion combinatoire au travers de l'adaptation dynamique des représentations des automates. Ce travail, en collaboration avec l'équipe MoVe du LIP6, s'intègrera au projet Spot.

1. L'air de rien, `http://publis.lrde.epita.fr/LrdeBulletin`.

Scribo : un projet pour la dématérialisation de documents.

par *Guillaume Lazzara*

De nos jours, les informations sont de plus en plus accessibles grâce aux réseaux et aux moteurs de recherche toujours plus performants. Cependant, une quantité importante d'information reste encore inexploitée à travers notamment toutes les archives et livres papier. Pour pallier ce problème, la tendance est à la numérisation et à la dématérialisation des documents. Le récent Grand Emprunt de la France portait d'ailleurs un volet spécifique sur ce sujet, ce qui ne fait qu'appuyer son importance aujourd'hui. Un autre problème qui surgit quand on parle de numérisation est l'indexation. Comment s'y retrouver dans des millions de documents numérisés ? Il est nécessaire d'exploiter correctement le contenu pour s'en servir pour l'indexation, mais pas n'importe comment ! C'est là que l'analyse sémantique intervient.

Grâce au pôle de compétitivité Systematic et son Groupe Thématique Logiciel Libre, en septembre 2009, le LRDE a rejoint le projet Scribo pour 2 ans et demi de travail en collaboration avec 8 autres partenaires : AFP, CEA, INRIA, Mandriva, Nuxeo, Proxem, Tagmatica et XWiki. Le but de ce projet était de fournir des outils d'indexation semi-automatique de documents en proposant notamment des solutions d'analyse sémantique.

Le rôle du LRDE dans ce projet était de fournir des outils pour la dématérialisation afin de pouvoir lancer une analyse sémantique à posteriori.

L'idée de la dématérialisation de documents consiste à numériser (scanner) et à analyser son contenu. Le but étant d'obtenir en bout de chaîne l'équivalent numérique en format PDF, XML, Open Document, ... En bref, un format dans lequel le texte est sélectionnable, les images sont bien découpées et la structure préservée.

A l'heure actuelle, il existe déjà quelques outils de référence dans le domaine, tel Abby FineReader. Ils sont plutôt adaptés à des documents variés et simples. De plus, ces solutions sont pour la plupart fermées et payantes. Une alternative open-source aurait donc toute sa place, ce qui justifiait d'autant plus notre participation.

Nous avons choisi de développer ces outils grâce à notre plateforme de traitement d'image Olena. Après 10 ans de développement, la bibliothèque de traitement d'image Milena était suffisamment stable pour s'attaquer à un tel projet. Nous avons mis au point des ensembles d'algorithmes de base et des chaînes de traitement afin de rendre le code modulable et réutilisable. Les résultats de ce développement étant probants, ce code a été intégré sous forme de module dans la plateforme Olena.

Le but de ce projet était également de fournir une solution de dématérialisation pour le grand public. Une interface a été développée à cet effet et fait également partie du module.

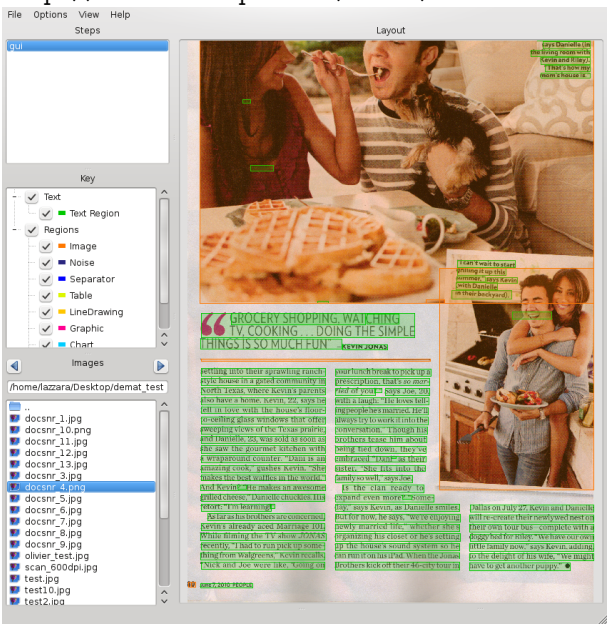
Grâce à ce projet, Olena est maintenant distribué sous format RPM dans la distribution Mandriva. Les chaînes d'extraction de texte dans les documents ont été également intégrées dans le bureau sémantique de KDE, Nepomuk.

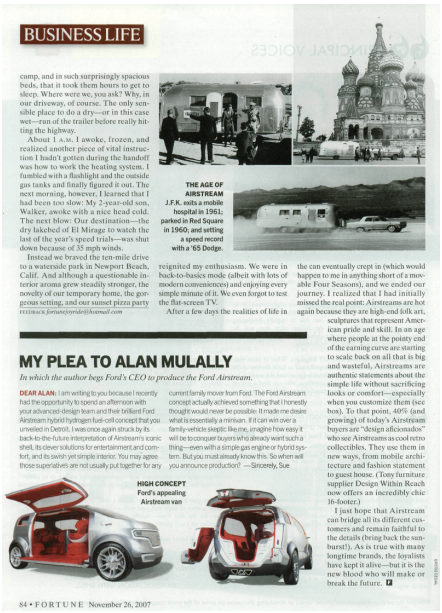
Enfin, nos traitements étant suffisamment robustes à certains types de documents, nous avons pu participer cette année à un concours de segmentation de documents type magazines et à un concours de segmentation de documents anciens. Nous avons atteint la 2ème place à ce dernier.

Scribo a été intégré à Olena et fait partie de la nouvelle version 2.0.

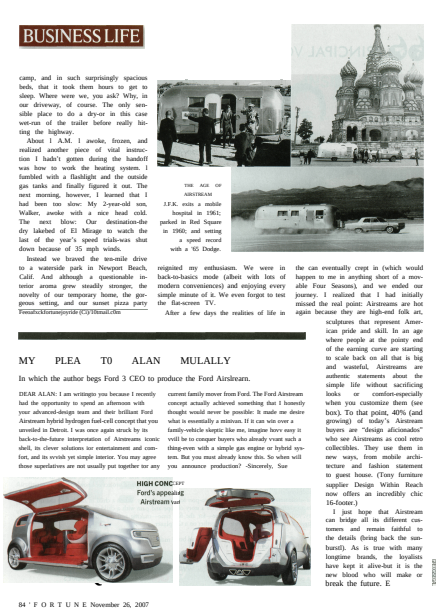
Pour ceux qui voudraient en savoir plus, un papier décrivant le projet a été accepté à la 11ème édition de ICDAR et est disponible sur notre site web.

Des démos en ligne sont aussi disponibles sur `http://www.lrde.epita.fr/Olena/Demos#Scribo`





(a) Image de document originale.



(b) Document reconstruit en PDF.

FIGURE 1 – Segmentation et reconstruction de document.

Présentation Javascript

par *Christopher Chedeau*

Les sites tels que Gmail, Facebook et Google Maps sont des exemples classiques d'utilisation de Javascript. Mais saviez-vous que l'interface de Windows 8 ou les extensions de Chrome et Firefox sont écrites en Javascript ? Ou qu'il est possible d'écrire des serveurs web en Javascript grâce à Node.js ?

Javascript est partout et pourtant, je me suis rendu compte en parlant autour de moi que personne ne connaissait réellement ce langage. C'est pourquoi je vous invite à une présentation de deux heures sur le sujet le **Mardi 25 Octobre en Amphi Master de 18h30 à 20h30**.

Javascript, le language

Pour commencer, un petit peu d'histoire. Brendan Eich raconte qu'il a pensé et implémenté le premier prototype de Javascript en 10 jours en 1995. En effet, Javascript est un langage qui contient un nombre extrêmement restreint de concepts. Cette idée provient du monde des langages fonctionnels tels que Lisp, Haskell ou Caml. Le génie de Javascript c'est d'avoir su s'écarter d'un modèle mathématique parfait au profit d'un confort d'utilisation pour le développeur.

Javascript a pour objectif d'être utilisé par le plus grand nombre de personnes. La syntaxe du langage a été fortement inspirée du C et ne contient aucune fantaisie. Cela rend le code source lisible et compréhensible par n'importe quel informaticien. Le language a été conçu pour exécuter un maximum de programmes, même mal formés. Par exemple, une heuristique va rajouter des point-virgules manquants. Au final, la barrière d'entrée au Javascript est très faible.

Lambda Fonctions et Objects

Javascript tire sa puissance de deux concepts fondamentaux : les *Lambda fonctions* et les *Objets*. La présentation a pour objectif principal de vous apprendre à manipuler ces deux outils. En guise d'introduction au langage, je montrerai comment reproduire des paradigmes de programmation connus, en particulier la Programmation Orientée Objet.

Le navigateur est un environnement hostile. Dans un site cohabitent une multitude de modules Javascript développés par des personnes différentes. On peut citer le site lui-même, les publicités, les commentaires, les statistiques, le bouton "like", etc. Nous verrons brièvement l'utilité des objets et des fonctions pour se placer dans l'un des trois points de vue suivants : être un citoyen respectueux, fortifier son code contre les attaquants ou au contraire s'amuser avec le code des autres.

Un langage dynamique

A l'école nous avons principalement étudié des langages de programmation statiques comme le C, C++ et Caml. Javascript quant à lui fait partie de la catégorie des langages dynamiques comme le PHP, Ruby ou Python. Les fonctionnalités de ces derniers ont pour objectif de simplifier la vie du développeur en s'éloignant des contraintes de la machine ou des théories mathématiques de typage. De ce fait, les langages dynamiques sont de plus en plus utilisés.

Nous étudierons lors du séminaire du 25 les changements apportés par cette nouvelle façon de penser. Par exemple, les chaînes de caractères sont utilisées de façon quasi systématique afin de faciliter le débuggage, les objets sont construits à la volée sans définir leur structure dans un fichier séparé pour gagner du temps, etc.

En bref

Les nouvelles publications

L'ensemble des publications du LRDE sont disponibles sur `http://publis.lrde.epita.fr/`.

Duret-Lutz, A., Klai, K., Poitrenaud, D., and Thierry-Mieg, Y.. Combining explicit and symbolic approaches for better on-the-fly LTL model checking. Technical Report 1106.5700, arXiv

> We present two new hybrid techniques that replace the synchronized product used in the automata-theoretic approach for LTL model checking. The proposed products are explicit graphs of aggregates (symbolic sets of states) that can be interpreted as Büchi automata. These hybrid approaches allow on one hand to use classical emptiness-check algorithms and build the graph on-the-fly, and on the other hand, to have a compact encoding of the state space thanks to the symbolic representation of the aggregates. The *Symbolic Observation Product* assumes a globally stuttering property (e.g., LTL$\setminus X$) to aggregate states. The *Self-Loop Aggregation Product* does not require the property to be globally stuttering (i.e., it can tackle full LTL), but dynamically detects and exploits a form of stuttering where possible. Our experiments show that these two variants, while incomparable with each other, can outperform other existing approaches.

Duret-Lutz, A.. LTL translation improvements in Spot. In *Proceedings of the 5th International Workshop on Verification and Evaluation of Computer and Communication Systems (VECoS'11)*, Electronic Workshops in Computing, Tunis, Tunisia. British Computer Society

> Spot is a library of model-checking algorithms. This paper focuses on the module translating LTL formulæ into automata. We discuss improvements that have been implemented in the last four years, we show how Spot's translation competes on various benchmarks, and we give some insight into its implementation.

Verna, D.. Towards LaTeX coding standards. In *TUGboat*

> LaTeX, en tant que simple système de macro-expansion, n'impose aucune forme de génie logiciel, structure de programme ou style de programmation. Contrairement à d'autres langages, l'idée d'un standard de programmation n'est pas tellement répandue dans le monde LaTeX, probablement parce que le travail collaboratif n'y est que peu représenté. Au fil des ans, un flot permanent d'expériences de développement a contribué à forger notre goût personnel en termes de style. Dans cet article, nous décrivons ce que nous pensons être de bonnes pratiques de programmation en LaTeX.

Verna, D.. Biological realms in computer science : the way you don't (want to) think about them. In *Onward ! 2011*

> En biologie, l'évolution est souvent perçue comme un processus de « bricolage », processus différent de ce que fait un ingénieur lorsqu'il planifie le développement de ses systèmes. Des études récentes ont cependant montré que même en biologie, il existe une part d'ingénierie. En tant qu'informaticiens, nous avons par contre beaucoup plus de mal à reconnaître qu'il existe aussi une grande part de bricolage dans ce que nous faisons, et que nos systèmes se comportent finalement de plus en plus comme des biotopes. Cet essai relate mon expérience personnelle dans ce domaine.

Soutenance de thèse de Roland Levillain

Roland Levillain soutiendra sa thèse de doctorat intitulée « Vers une architecture logicielle pour le traitement d'images générique » le mardi 15 novembre 2011. Le lieu de la soutenance sera communiqué ultérieurement.