

# Rapport de stage M1 au Trinity College de Dublin

Grégoire Détrez

Du 2 Juillet au 31 Août 2007

## 1 Déroulement du stage

Je cherchais un stage dans un laboratoire de linguistique informatique. J'ai fait une demande à Aachen (j'y avais entendu parler de linguistique lors de la visite) mais ils n'y avait personne de disponible pour encadrer un stagiaire. J'ai donc chercher ailleurs. J'ai trouvé le laboratoire de linguistique informatique du Trinity college intéressant et, après avoir pris contact avec Carl Vogel, celui-ci m'a proposé 5 sujets de stage possibles. J'en ai sélectionné deux avec mon tuteur à l'ENS et c'est Carl Vogel qui a défini le sujet final.

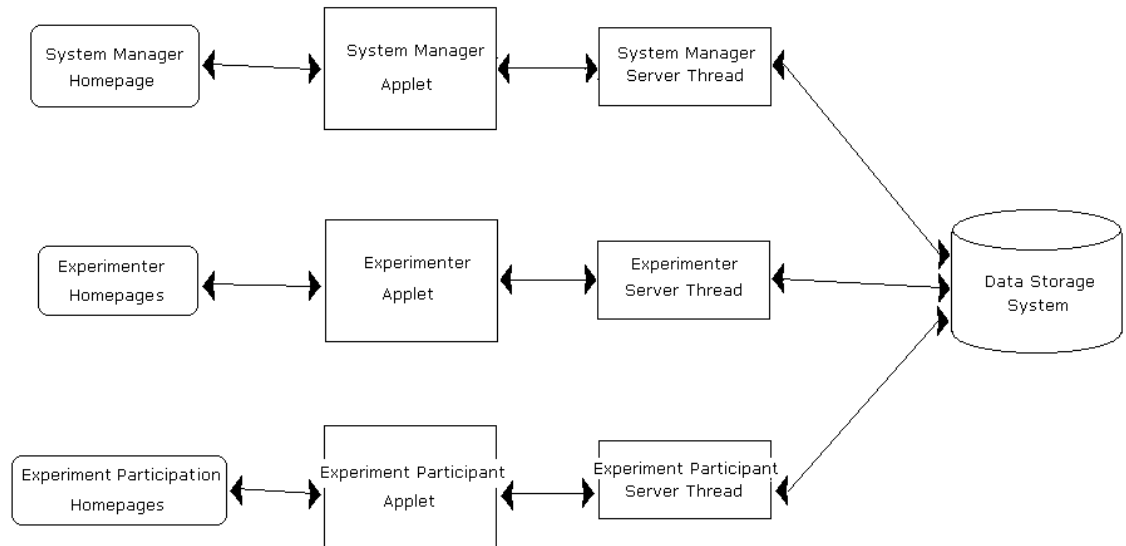
## 2 Dublin et le Trinity College

Le Trinity college est la plus ancienne université d'Irlande. Très agréable, la plus part des bâtiments sont d'anciennes construction, on y trouve notamment la magnifique vielle bibliothèque (maintenant devenue un musée.)

Très touristique l'été, la capitale est tout de même agréable. Il y a beaucoup d'animations.

## 3 Système d'expérimentations

### 3.1 conception globale



Le système était en fait divisé en trois systèmes pratiquement indépendants. Le premier permet à l'administrateur du site de créer et de modifier les utilisateurs. Le second sert aux expérimentateurs à créer, modifier et analyser les expérimentations. Le dernier distribue les expérimentations aux sujets et collecte les résultats. La redondance était importante entre ces trois systèmes.

### 3.2 protocole de communication

Le protocole était assez compliqué. En fait, à chaque "événement" au niveau du client correspondait une instruction du protocole. Par exemple, le client envoyait l'instruction "LOGIN\_OK" pour indiquer que l'utilisateur était correctement identifié et qu'il allait afficher la page d'accueil. Le serveur répondait en envoyant la liste des expériences disponibles pour l'utilisateur. Il semblait plus logique d'avoir une instruction "LIST experiment" qui reflétait mieux le type de réponse attendu.

De plus, le serveur et le client maintenaient chacun une copie des objets utilisés. Lors de l'envoi d'un document au client, le serveur créait lui aussi une instance de l'objet sauvé dans le document et en cas de modifications par le client le serveur commençait par mettre sa copie à jour avant de l'enregistrer sur le disque. Cette duplication des objets rendaient le protocole difficile à

maintenir.

Il y avait des erreurs de conceptions assez étranges. Par exemple, le serveur n'effectuait pas la vérification du mot de passe pour modifier les données mais envoyait ce mot de passe au client et c'était à ce dernier de vérifier la correspondance!

La première chose que j'ai faite à donc été de simplifier ce protocole. Le protocole ne s'occupe plus que de servir les fichiers et de vérifier les mots de passe, sans s'occuper de ce qu'il se passe chez le client.

### 3.3 architecture des GUIs

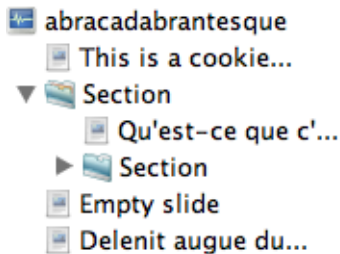
Les applets java étaient écrites comme une application simple, un prototype sans doute, auquel on aurait petit à petit ajouté des fonctionnalités, ce qui aboutit au syndrome classique des fonctions de 3000 lignes.

De plus, elles étaient programmées de manière très peu orientée objet. Une méthode principale traitait l'action en cours, n'importe laquelle, et une seule classe s'occupait de toute l'applet.

Le but de mon stage était d'améliorer l'éditeur pour permettre : - de réorganiser les pages - d'ajouter des images directement depuis l'applet (l'ancien système ne permettait que d'utiliser des images qui se trouvaient déjà sur le serveur.)

Pour cela, j'ai changé le format de fichier sous lequel étaient stockées les expériences. Il s'agissait d'un format en texte qui s'appuyait sur des balises du type latex. Le nouveau format s'appuie sur XML, ce qui permet d'utiliser les nombreuses bibliothèques existantes pour lire les fichiers. Pour insérer des images dans ces fichiers non binaires, j'ai converti les fichiers en base64. Ce système est utilisé dans les logiciels de courriels pour transmettre des fichiers binaires dans les courriers électroniques.

Pour ce qui est de l'amélioration de l'éditeur, j'ai ajouté une vue "en arbre" qui permet de voir la totalité de la structure de l'expérience et chaque groupe d'éléments peut être réorganisé grâce à une liste avec glisser-déposer.



## 4 Conclusion

Le stage concernait seulement de l'implémentation. Le travail consistait entre autre à "découvrir" du code écrit par quelqu'un d'autre. J'aurais personnellement préféré faire plus de linguistique.

## Références

[Graham2005] Yvette Graham, *Services for Experimentation in the Human Sciences : An Online Experimentation Tool*, M.Sc. Computational Linguistics, Thesis, October 2005

## 5 Annexes

TAB. 1 – Codage base64

Valeur	Codage	Valeur	Codage	Valeur	Codage	Valeur	Codage
0	A	17	R	34	i	51	z
1	B	18	S	35	j	52	0
2	C	19	T	36	k	53	1
3	D	20	U	37	l	54	2
4	E	21	V	38	m	55	3
5	F	22	W	39	n	56	4
6	G	23	X	40	o	57	5
7	H	24	Y	41	p	58	6
8	I	25	Z	42	q	59	7
9	J	26	a	43	r	60	8
10	K	27	b	44	s	61	9
11	L	28	c	45	t	62	+
12	M	29	d	46	u	63	/
13	N	30	e	47	v		
14	O	31	f	48	w	(complément)	=
15	P	32	g	49	x		
16	Q	33	h	50	y		

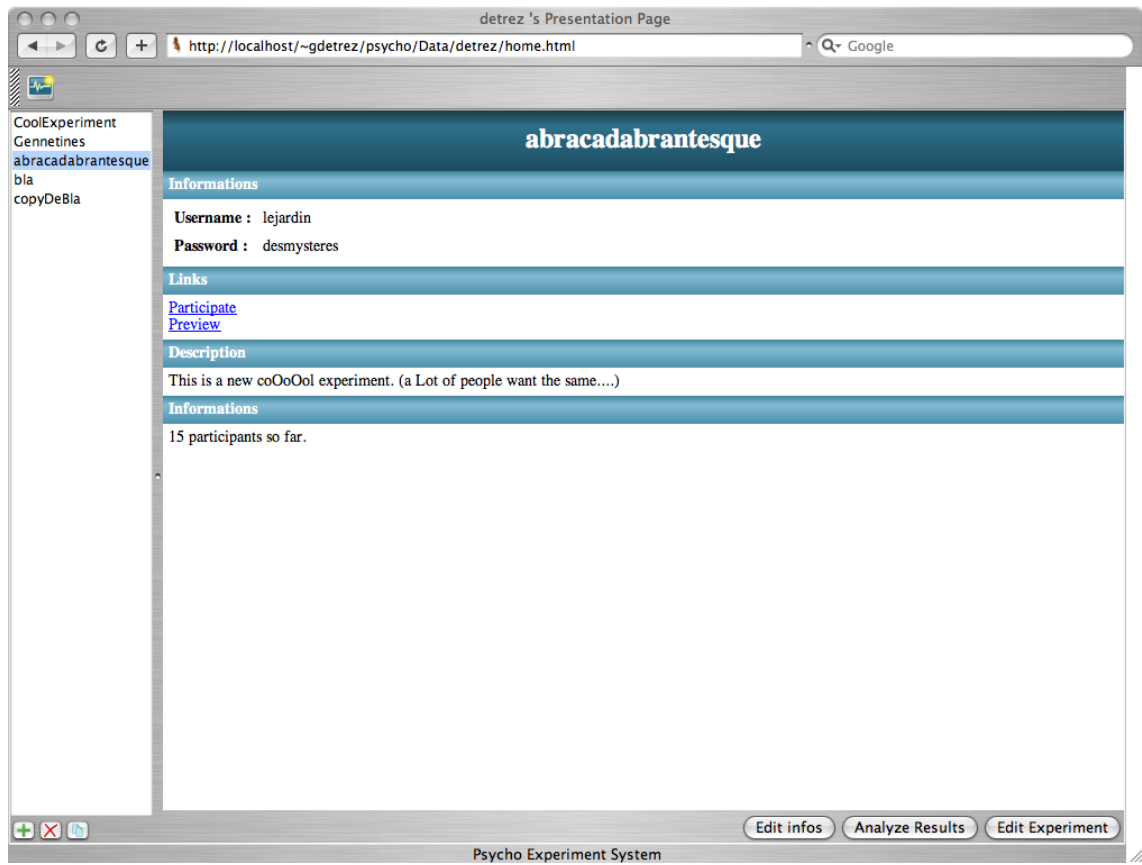


FIG. 1 – Liste d'expérimentations

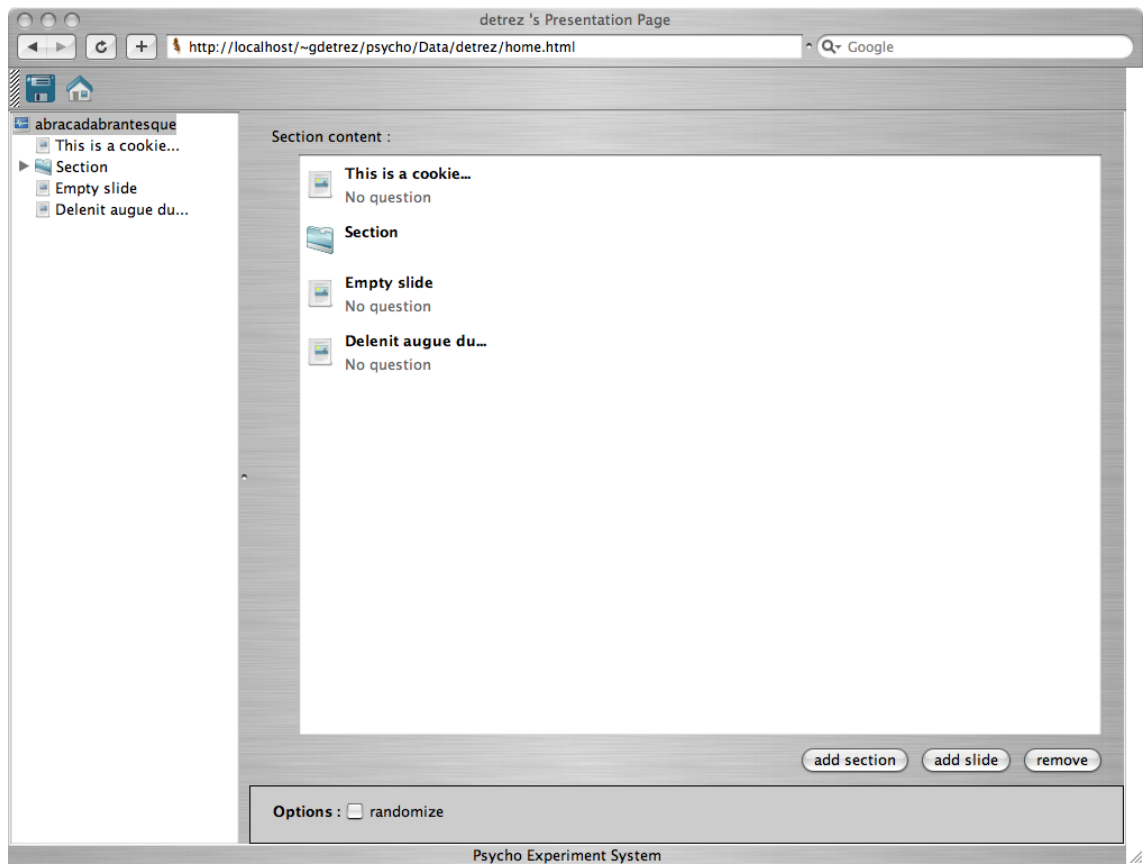


FIG. 2 – Édition