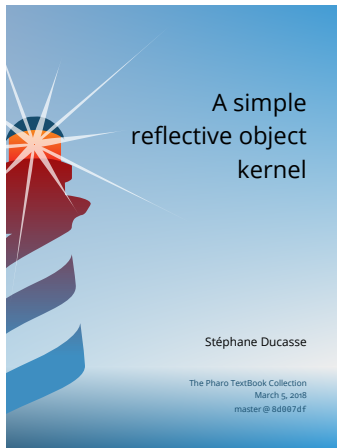


J'ai découvert l'article sur ObjVlisp en 1993 à la fin de mon master et du cours de Mireille (Blay-Fornarino) sur la programmation réflexive et cela m'a tout de suite éclairé sur tous les points que je n'avais pas compris mais que je me gardais bien d'exposer. Après la lecture de l'article, j'avais été pris d'une pulsion pour l'implémenter, et hop une implémentation en Lisp avec un bootstrap tout mignon. Quel plaisir d'implémenter ce micro noyau.

J'ai immédiatement dit à Mireille qu'elle devrait l'enseigner tellement c'était une clef conceptuelle. Mireille m'a alors dit de faire le cours. J'ai donc depuis lors enseigné ObjVlisp quasiment chaque année avec maintenant bien sûr une implémentation en Pharo et un petit livre bientôt publié.



J'ai compris aussi la remarque de (Roger) Rousseau, mon prof d'Eiffel à propos de la représentation des variables partagées. Oui, pas besoin de définir une nouvelle métaclasse et de mixer les niveaux, il suffit comme dans Smalltalk que les classes aient des ClassVariables.

Objvlist m'a beaucoup inspiré tout comme le noyau de Smalltalk-78. MetaclassTalk, NeoClassTalk et la composition de métaclasses ont été des travaux importants pour moi quand j'ai quitté CLOS pour sombrer dans Smalltalk. Nous nous sommes bien amusés à penser à des classes auto-décrites.

Dès 2002, je rêvais d'un noyau Smalltalk bootstrappable depuis Smalltalk lui-même et pas un bidule que l'on crée en copiant des objets dans un autre fichier comme le systemTracer. Bien des années plus tard en 2012, lorsque j'ai proposé à Guillermo Polito de travailler sur la construction de noyaux de langages dédiés et le bootstrap de Pharo, je ne pensais pas que nous pourrions créer des images de 30k avec PharoCandle un noyau complet en 80K. Je ne pensais pas que nous arriverions à mettre une bombe dans la métaphore des images Smalltalk que l'on tripote allègrement afin d'injecter un processus de GL et Guillermo nous a montré que c'était possible. Jubilatoire! Pharo à ce jour est bootstrappé depuis lui-même à chaque commit : Un nouveau noyau tout beau tout propre. Cela a permis à des sociétés de faire tourner un noyau Pharo de 200k dans une machine virtuelle javascript en utilisant SqueakJS. L'histoire continue avec Luc Fabresse, nous voulons nous remettre à l'ouvrage en revisitant la conception, production et le debuggage de tels microsystemes (je me rappelle debugger des micro noyaux en écoutant si Bip était produit - oui un très rudimentaire printf.

Donc merci Pierre pour cet article si important pour moi.

Pour le modèle je l'aurais appelé Ouroboros tellement le serpent se mangeant la queue me fait penser à un système réflexif.

Stéphane Ducasse