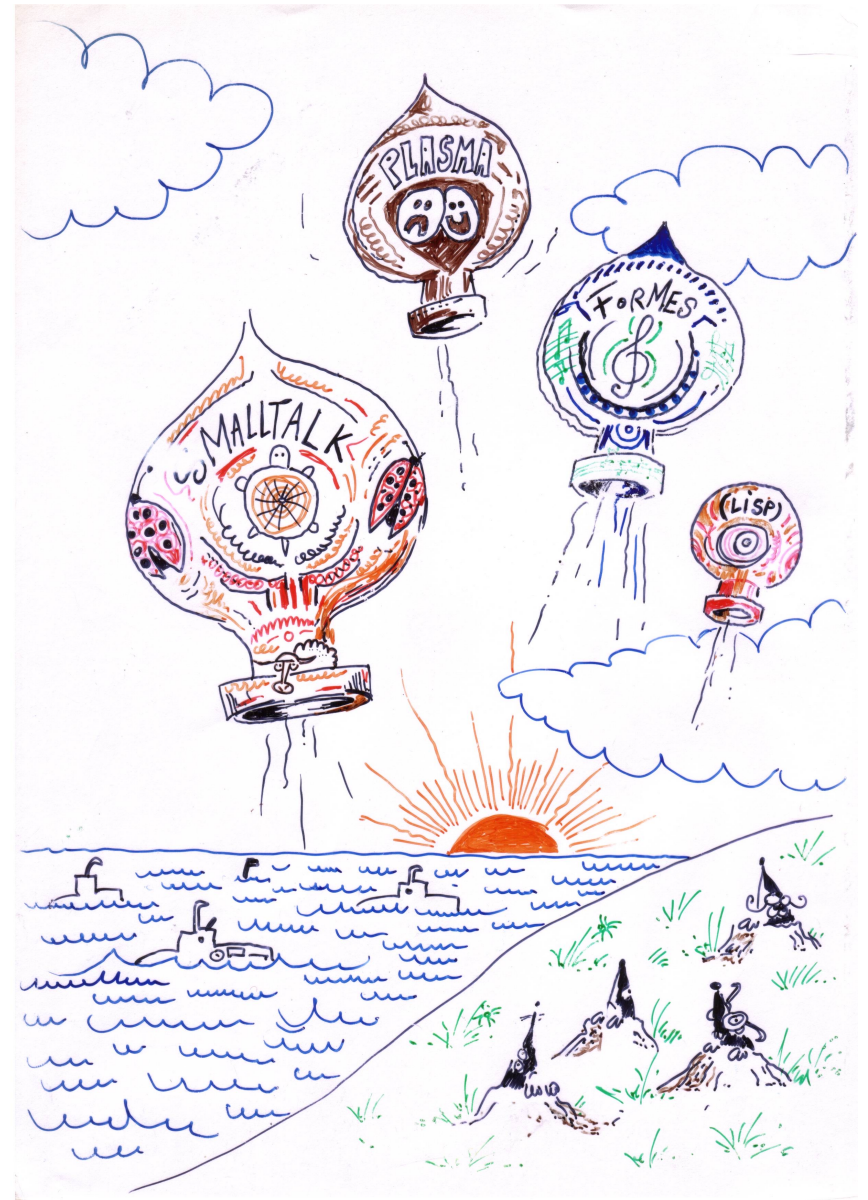


JPB et les année 80.
Le surf sur la vague des objets

Pierre Cointe
Journée JPB
Jussieu, tour 25-26, le 13 avril 2023



1981-1991. C'était l'époque

De l'Informatique musicale

Des langages à objets

- Instanciation et héritage
- Classes versus prototypes

Des architectures réflexives

- Classes comme objets de première classe
- Protocoles de méta-objets
- ObjVlisp et Classtalk

Des objets concurrents

- Acteurs comme objets actifs
- ABCL/1 et Actalk

Des naissances d'ECOOP et d'OOPSLA

Des séjours longs à Tokyo et Los Angeles

- A. Yonezawa (TIT)
- L. Gasser (USC)



1982. DEA LAP : IRCAM et LITP

Analyse et implémentation du langage Formes

HUMANO HUMANO - HUMANO - HUMAI

IRCAM: L'IA SUR LE MODE MINEUR

"Nous sommes les luthiers du XXI^{ème} siècle", affirment avec quelque coquetterie les chercheurs de l'IRCAM. Vrai et faux : la musique contemporaine s'intéresse aujourd'hui autant à découvrir de nouveaux modes de "composition" que de nouveaux "instruments". FORMES, un environnement de programmation orientée objets, fait les deux à la fois, si la musique (vraiment) contemporaine venait tout juste de naître ?

alors existant – sont peut adaptés à l'élaboration et à l'utilisation des systèmes complexes de l'informatique musicale.

En lançant il y a bientôt deux ans à l'IRCAM le système FORMES, Xavier RODET et Pierre COINTE ont radicalement modifié les données du problème. La Synthèse et Composition Musicale par ordinateur – la SCM – doit en effet répondre, pour être fiable, à plusieurs impératifs simples à formuler, mais passablement compliqués à mettre en œuvre.

D'abord, un programme de SCM doit pouvoir être très souvent et très vite réécrit, testé, modifié pour répondre aux intentions finales du compositeur qui reste, en théorie, totalement responsable de ses choix sonores ou structurels. De plus, le système réclame une **interactivité totale** et rapide pour **canaliser** en temps réel les synthétiseurs auxquels il est connecté. Enfin, **il** les informaticiens, ni surtout les compositeurs ne doivent être freinés dans leur démarche créatrice par des programmes trop **dogmatiques**. Comme le fait remarquer Jacques DUTHEN, technicien du logiciel à l'IRCAM et lui-même pourvu d'une solide culture musicale : **"ils ont des années de Conservatoire derrière eux, ils n'ont pas envie de passer dix ans à apprendre la programmation"**.

La réponse à ces exigences – et à quelques autres, plus spécifiques, comme la non-dépendance à une technique d'analyse musicale quelconque, la compatibilité aux concepts universaux d'acoustique ou la stricte hiérarchisation des modèles –, on la trouve dans les "langages orientés objets", les LOO, un mode de programmation qui intéresse de plus en plus le monde de l'intelligence artificielle.

Processus-parent et processus-enfant

De manière générale, la programmation orientée objet (POO) définit, à la fois pour le code et pour les données, un certain nombre d'"objets" capables, sous l'influence de messages extérieurs adaptés, de réaliser des opérations – des "méthodes" – sur leurs données locales, appelées "accommodances". Bien sûr, une méthode peut à son tour émettre des messages vers d'autres objets. La philosophie générale de la POO est donc extrêmement décentralisée : chaque objet, hyper-spécialisé, ne tra-

vaillera que sur les données qui lui appartiennent en propre, et ne s'intéresse en aucune manière au traitement qui s'opère dans les autres objets auxquels il envoie messages et données. En outre, le programmeur peut définir une "généalogie", un "objet-fils" étant alors généré par un "objet-père" et héritant tout ou partie des caractéristiques de ses "ancêtres". A ce mode de fonctionnement particulier correspondent deux avantages essentiels : les programmes restent extrêmement concis, et les objets suffisamment généraux pour être réutilisables dans des circonstances différentes.

Le système FORMES élaboré à l'IRCAM recoupe les grandes orientations des LOO. Des processus y regroupent des règles de calcul, un moniteur temporel, un environnement de variables locales et une descendance plus ou moins fournie. Activés par des messages déterminés, les processus assurent, dans un laps de temps défini, le calcul d'une caractéristique musicale donnée (phrasé, vibrato, timbre, intensité). Le moniteur assure l'activité et l'ordonnement exacts des processus, eux-mêmes organisés en généalogie selon un schéma

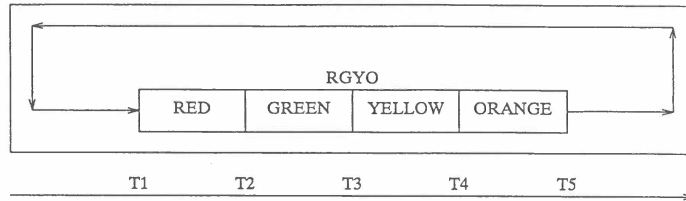


Sur Macintosh

FORMES a été élaboré en VLISP, initialement sur un DEC PDP-10, puis sur un DEC VAX-11/780. Il tourne actuellement sur Macintosh configuré de la façon suivante : 512 K RAM, un lecteur de disquettes externe, une interface MIDI connectée sur le port "série" standard du Macintosh. Le choix du Macintosh permet également de disposer d'une interface "agréable" de programmation, la définition des fonctions d'enveloppe pour les sons et la création d'ensembles de paramètres pouvant par exemple se réaliser grâce à la souris. Les langages utilisés sont VLISP et LE LISP. Mais on pourrait parfaitement produire d'autres versions de FORMES pour d'autres machines et dans d'autres langages.

Le système Formes élaboré à l'IRCAM recoupe les grandes orientations des LO. Des processus y regroupent des règles de calcul, un moniteur temporel, un environnement de variables locales et une descendance plus ou moins fournie. Activés par des messages déterminés, les processus assurent dans un laps de temps défini le calcul d'une caractéristique musicale donnée (phrasé, vibrato, timbre, intensité). Le moniteur assure l'activité et l'ordonnement des processus eux même organisés en généalogie ...

IA Cognitech 1986



The process **RGYO** coupled with the monitor **circ-node** schedules the repetitive sequence RED to GREEN, GREEN to YELLOW, YELLOW to ORANGE, ORANGE to RED, RED to GREEN, and so on...

Here is the complete FORMES program realizing the rocking trajectory:

```
(dp RGYO
  monitor:      circ-node
  sons:         (RED GREEN YELLOW ORANGE)
  first-time:   ((send 'clock '?<- 'quantum 0.1))
  exit:         (> time #@etime)
  env:          (duration 60.) )

(dp RED
  monitor:      leaf
  each-time:    ((setq f1 (trajectory #@father #@direction #@α #@φ #@ω))
                 (send fself 'suspend))
  env:          (direction t α 4. φ π/2 ω 7.) )

(send 'RED 'new 'GREEN 'env: '(direction nil ω 15.))
(send 'GREEN 'new 'YELLOW 'env: '(φ -π/2 ω 30.))
(send 'YELLOW 'new 'ORANGE 'env: '(direction t ω 60.))
```

- **RGYO** schedules the change of trajectory by giving alternatively the control to each of its sons. Because the associated genealogical tree is circular, this process cannot finish with its last son, so we have to designate the **RGYO** termination by the *exit*: stop condition. The **RGYO** end occurs when the clock outstrips its *etime*, i.e., when ($> \text{time } \#@\text{etime}$) is true.

- The duration of the **RED** process is mapped with the quantum to associate the duration " $T_{i+1}-T_i$ " with the duration between two ticks of the clock. The message *suspend* allows the suspension of the receiver process at its first (and last) tick.
- **GREEN**, **YELLOW**, and **ORANGE** are *differentially* derived from the **RED** definition.

The following figure gives the temporal evolution of the *f1* register after an activation of **RGYO**:

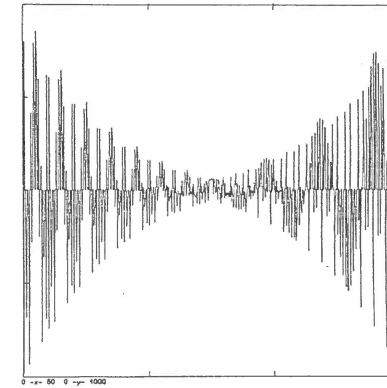


Figure 5. CHREODE, fundamental frequency: *f1* (time)

5. GOD Process or a First Step Toward a Meta-FORMES

This section presents the activation protocol of a process **P** and explains the behavior of the only predefined process: **GOD**.

When **P** receives the message *run*, the `#:object:process:run` function is applied (the *run* function is interned in the `#:object:process` package):

1984. Conference on Lisp and Functionnal Programming

MONDAY AFTERNOON, AUGUST 6

02:10 PM Session 3

Chairman: John Williams (International Business Machines)

- "Listlessness is Better than Laziness" Philip Wadler (Oxford University)
- "Stream Processing" Allen Goldberg (University of California, Santa Cruz, and Kestrel Institute) and Robert Paige (Rutgers University)
- "Rewriting Systems on FP Expressions that Reduce the Number of Sequences They Yield" Francoise Bellegarde (Centre de Recherche en Informatique de Nancy)
- "Schema Recognition for Program Transformations" John S. Givler and Richard B. Kieburtz (Oregon Graduate Center)

03:50 PM Coffee break

04:15 PM Session 4

Chairman: Guy L. Steele Jr. (Tartan Laboratories)

- "Formes: an Object and Time Oriented System for Music Composition and Synthesis" Pieere Cointe and Xavier Rodet (Institut de Recherche et Coordination en Acoustique/Musique)
- "Arctic: A Functional Language for Real-Time Control" Roger B. Dannenberg (Carnegie-Mellon University)
- "muFP, A Language for VLSI Design" Mary Sheeran (Oxford University)

05:30 PM Break for dinner

MONDAY EVENING, AUGUST 6

08:00 PM Session 5

Chairman: Richard P. Gabriel (Stanford University)

- "LE LISP, a Portable and Efficient LISP System" Jerome Chailloux (INRIA), Matthieu Devin (Ecole des Mines de Paris), and Jean-Marie Hullot (INRIA)
- "DIALISP - A LISP Machine" G. Stefan, A. Paun, A. Birnbaum, and V. Bistriceanu (Functional Institute of Bucharest)
- "Recent Developments in ISI-Interlisp" Raymond L. Bates, David Dyer, and Mark Feber (University of Southern California Information Sciences Institute)
- "TAO: A Fast Interpreter-Centered Lisp System on Lisp Machine ELIS" Hiroshi G. Okuno, Ikuo Takeuchi, Nobuyasu Osato, Yasushi Hibino, and Kazufumi Watanabe (Nippon Telegraph and Telephone Public Corporation)
- "The Design of an Instruction Set for Common Lisp" Skef Wholey and Scott E. Fahlman (Carnegie-Mellon University)

- Formes : an Object and Time Oriented System for Music Composition and Synthesis
- Aide inestimable de Jean-Pierre pour la relecture et l'envoi du papier
- Même session présentation d'Artic (Dannenberg CMU)
- Présentation de LE LISP dans la session suivante

The Formes System: A Musical Application of Object-Oriented Concurrent Programming

Pierre Cointe
Jean-Pierre Briot
Bernard Serpette

This paper presents the FORMES system developed at IRCAM to deal with the complexity of musical representation and used to drive a general sound synthesizer.

We first focus on the original concepts induced by musical applications. Object-Oriented Programming matches a subset of the numerous requirements of Computer Music Composition, but we need to extend this metaphor towards the time component in order to describe and control the temporal structures of music. Therefore we introduce the monitor concept which supports different schedulings of hierarchical processes, and extends the class concept to the field of concurrent (musical) structures.

A short tutorial of the system describes its use through some musical examples. The FORMES virtual machine implemented in LISP is also presented.

1. Introduction

The FORMES system is a programming environment initiated at IRCAM¹ four years ago. It aims at music composition and synthesis (MCS) by computer. The goal is to address the complexity of musical representation and to provide for resident or guest composers - some of them with little or no computer expertise - a powerful and friendly environment for music research and production.

FORMES provides a musical framework modeling the results obtained in signal processing and synthesis research into sound material organized as blocks of knowledge

¹ L'Institut de Recherche et Coordination Acoustique/Musique is "conducted" by Pierre Boulez.

called *processes*. These processes must be "easily" composed to elaborate the musical score that will be executed by various synthesizers (e.g., the CHANT program [Rodet et al. 1984], the MIDI interface or the 4X real-time processor [Koechlin et al. 1985]).

The implementation of FORMES uses the techniques of object-oriented systems and develops them in a context of *concurrent musical programming*.

To date, a number of composers have used it for their work (e.g., J. Harvey, P. Manoury, K. Saariaho, M. Stroppa, ...). As an example, CHREODE [Barrière 1984], the first piece composed with FORMES by J. B. Barrière, won the "International Bourges Festival" award in 1983.

First we wish to demonstrate how exciting it is to test object-oriented methodology in a context of musical research with the support of famous composers. Then we present some examples of FORMES scores and introduce the implementation choices used to build such a complex musical system on LISP foundations.

More precisely: (1) we discuss the qualities of object-oriented programming for the musical domain, and we justify our choice for developing an object level upon a LISP kernel; (2) we introduce the major FORMES classes of objects, which allow a precise control of time and a simplified data-flow: we call them *process*, *monitor*, *clock*, and *calculation tree*; (3) we detail a tutorial including a meta definition of the GOD process, explaining the connection between the different kinds of objects and giving various examples of FORMES predefined musical structures; (4) we define the virtual machine which receives *message-passing* expressions and macro-expands them toward optimized LISP S-expressions operating upon the internal representation associated with each class of objects. This machine is built to be immediately installed on any dynamically scoped LISP, the standard IRCAM version being the Le_Lisp system [Chailloux et al. 1984] of INRIA; (5) we conclude with a criticism of some conceptual choices and present the future developments of the FORMES system.

2. Object-Oriented Metaphor Matches a Subset of MCS Requirements

2.1. Requirements of MCS

We will take here the initial requirements as expressed by X. Rodet at the beginning of the project (August 1981).

In MCS, the aim of a musician is to capture some musical image. A MCS program is an attempt to find and realize this image - using a model that implements our knowledge about sound production and perception - within a particular compositional context. Consequently FORMES must provide a framework to manipulate and integrate these models as building blocks or objects. These models are characterized by their

La coopération avec les compositeurs

- Tod Machover
- Jonathan Harvey
- Chris Chafe
- Marco Stroppa

- Jean-Baptiste Barrière
 - Chreode1 Festival de Bourges 83
- Kaija Saariaho

- Philippe Manoury (interface 4X)

Voyage au cœur de l'IA. Libération 2017

Gérard Assayag

En 1973 Patrick Greussay, chercheur à Vincennes et gourou de Lisp en France, expose dans sa thèse des méthodes d'intelligence artificielle au service de l'analyse d'œuvres musicales. Greussay fréquente l'Ircam dès sa création quelques années plus tard et y anime un séminaire d'informatique auquel Pierre Boulez ne dédaigne pas de participer.

Dès la fin des années soixante-dix et au début des années quatre-vingt, les chercheurs en informatique musicale de l'université de Stanford aux Etats unis et de l'Ircam en France s'emparent de ce langage pour développer des environnements créatifs pour le compositeur. FORMES écrit par Pierre Cointe, ainsi que CRIME par Gérard Assayag et Claudy Malherbe constituent à l'Ircam les premiers environnements Lisp d'assistance à l'invention et à l'écriture musicale, qui connaîtront une nombreuse descendance tels que Esquisse, Patchwork ou OpenMusic; ce dernier toujours utilisé et en évolution. Ces environnements dans un langage de très haut niveau favorisant la modélisation de structures symboliques donneront lieu à toutes sortes d'expériences d'IA autour des grammaires génératives, de la programmation par contraintes, de la représentation des connaissances musicales.

Parallèlement, un autre fil se déroule autour du langage d'IA PROLOG inventé par Alain Comerauer à Marseille en 1972 et basé sur la logique formelle. Le langage visuel CARLA, écrit par Francis Courtot au début des années 90 à l'Ircam, était basé sur Prolog et permettait de concevoir des séquences musicales comme des raisonnements résultant de contraintes logiques.



19/10/1983. Le Cap d'Agde

Première journée langages orientés objet

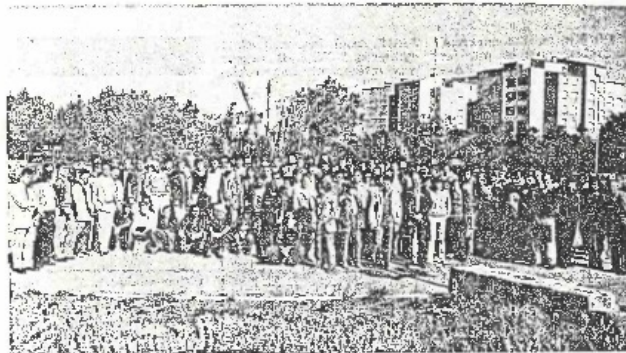
L'instanciation dans les langages objets

S O M M A I R E	
	Page
- PRESENTATION.....	1
- Les Objets dans Logo par Isabelle BOGNE, LITP, Paris.....	2
- Coproducts Smaltalk, Penser Smaltalk, par Pierre COINTE, IRCAM et LITP, Paris.....	13
- Mylsee : Un langage acteur pour le développement d'ateliers logiciels, par Yvon ALMET, CNRS LAMNIN 2.....	53
- Acteurs et résolution logique par Patrick RAHLE et Françoise CARRE, ISI-ENSEEINT, Toulouse.....	68
- A propos de CLU par Pierre LESCANDRE, CRIN, Nancy.....	86
- Kool : représentation des connaissances par Patrick ALBERT Centre de Recherche RMI, Paris.....	88
- Communications inter-processus par variables partagées par Bernard SERPENT, LITP et IRCAM, Paris.....	92
- Programmation par acteurs par Jean Louis GURJAN et Danielle JULIEN, Laboratoire 157, IRE, Toulouse.....	115
- Une mise en oeuvre de Smaltalk-80 par Jean BERTVIN et Bernard GUTHRIE, IRE et LITP, Irest.....	130
- The Contributions of an object-oriented approach to the programming of distributed applications par Marc ZWARTRO, CNRS, Paris.....	149
- Les langages objet dans la représentation des connaissances : application au langage MELING par Jacques FERBA, Paris VI, Paris.....	160
- L'instanciation dans les langages objets par Jean Pierre BRIOT LITP et IRCAM, Paris.....	173
- Coex : un environnement de programmation multifonctionnel par Jean Marie HALLLOT, IREIA, Roubaix.....	210
- Liste des participants.....	

Notes de la Rédaction :

Les numéros 35 à 37 de BIERE contiennent les comptes-rendus des Journées BIERE et des Journées d'étude sur les langages orientés Objet, organisées au Cap d'Agde en Octobre 1983. Le numéro 37 est publié avant le numéro double 36-36 pour des raisons techniques.

Les Journées BIERE vues par un journaliste du *Hercule-Tribune* :



Le Génie Logiciel au Palais des Congrès

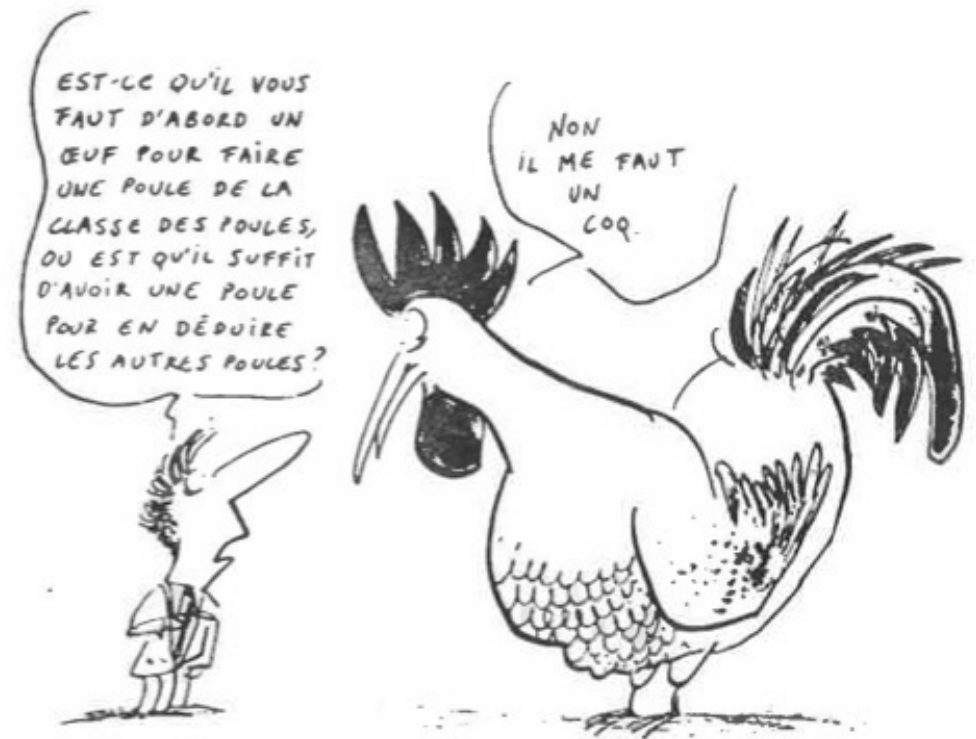
Du 17 au 19 octobre, les Journées Bigre 83 ont réuni au Centre des Congrès du Cap d'Agde environ 200 informaticiens, chercheurs et industriels qui se sont penchés sur ce problème crucial de l'informatique qu'est le génie logiciel. Une soixantaine d'exposés ont traité de divers domaines aussi importants que la conception, la spécification, les outils d'aide à la programmation, la validation et les ateliers de génie logiciel.

15/12/1984. Soutenance thèse JPB
Instanciation et héritage dans les
langages objets.

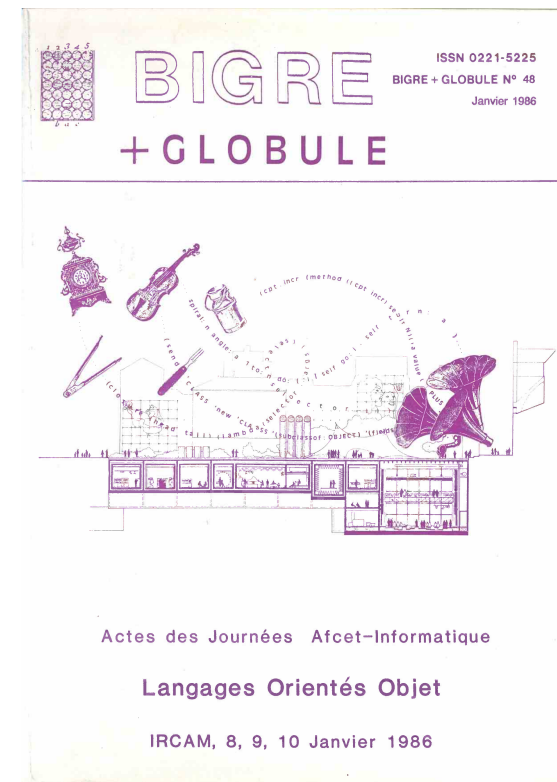
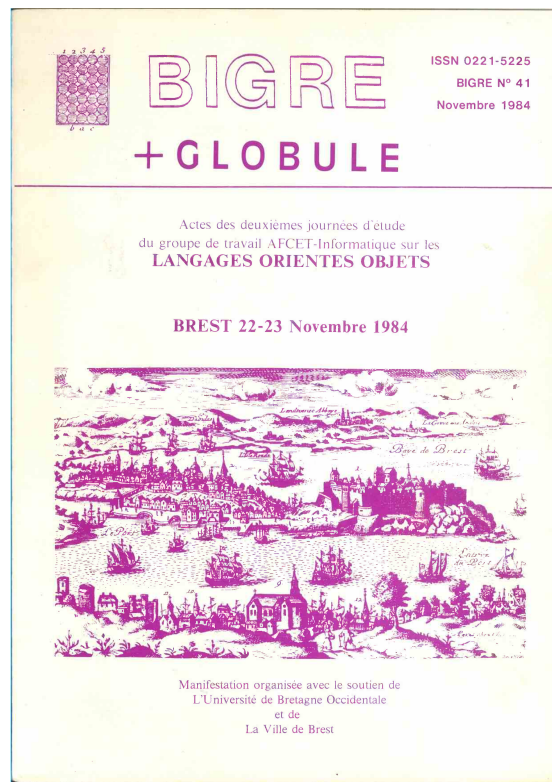
Pour reprendre la distinction sartrienne : « l'essence précède l'existence » dans le modèle objet, et « l'existence précède l'essence » dans le modèle acteur. Ou encore, le modèle objet est platonicien, tandis que le modèle acteur est existentialiste....

Or, l'expérience du développement de Formes à l'IRCAM a mis en évidence le besoin de faire appel aux deux types de mécanismes simultanément. Pour créer les objets très complexes que sont les processus musicaux, il faut pouvoir aussi bien les spécifier a priori que les reproduire par copie.

Extrait du rapport de thèse de Jean-François Perrot



Après Agde, Brest en novembre 84 et IRCAM en janvier 86



Intégration de Smalltalk dans le projet AGLAE

Réécriture et récursion dans une fermeture ;
étude sur un Lisp à liaison superficielle,
application aux objets

BIGRE+GLOBULE n°41, Novembre 1984

SOMMAIRE

Actes des Journées « Langages Orientés Objets »

• J. Bezin et P. Cointe Présentation des Journées	2
• P. Gressat Présentation de SCHEME à partir d'un exemple	5
• Y. Noyelle Un système PROLOG écrit en SMALLTALK-80	20
• I. Borne Présentation du système MICRO-SMALLTALK	30
• J.P. Briot Intégration de SMALLTALK dans le projet AGLAE	52
• P. Sinz Quelques sur le langage C++ , ou les drivers Unix vus comme des objets	73
• J.F. Cadelou, J. Collet et J.F. Cloarec Modélisation de spécifications sous forme d'objets et d'actions	81
• C. Roche Caractéristiques de LRO (Langage de représentation d'objets)	101
• M. Devin, M. Madeline et A. Ressouche Applications du langage objet CEIX à la construction de programmes sous forme de machines virtuelles	120
• B. Pottier Première évaluation d'une machine virtuelle SMALLTALK-80 sur SM90	141
• P. Cointe FORMES par l'exemple	152
• J. Bezin Simulation et langages orientés objets	194
• P. Gautron Une application multiphonique du système multi-tâches en classes C	212
• S. Habert Le Class pré-processeur C : une application ludique	228
• B. Serpette Une implémentation de PIVE en ObjYlisp	236
• Y. Autret Une présentation d'Ulysse	250
• M.J. Trebaul L'interface utilisateur en SMALLTALK-80	259
• J. Ferber Quelques aspects du caractère self réflexif du langage MERING	277

Bulletin publié avec le concours de l'AFCT-INFORMATIQUE



SOMMAIRE

• (P. Cointe, J. Bezin) Présentation des journées	1
• (Ch. Renot, Y. Casseau, Ch. Phavong) LORÉ : Un Langage Objet Relationnel et Ensembliste	3
• (J.M. Goetz, C. Rocha) Implémentation du Langage de Représentation d'Objets : LRO2	14
• (Shapiro, S. Hubert) Un système d'exploitation orienté objets pour SOMM	21
• (O. Danvy) Un développement original de C vers les acteurs vs les processus : le modèle Xinu	29
• (P. Deuzan) Alternative Design Styles Using the Smalltalk-80 System (résumé étendu)	36
• (S. Cook, G. Roberts) A Compile-Time Type-Checking System for Smalltalk-80	40
• (K. Doyle, B. Haynes, M. Lentschner, L. Rosenstein) An Object Oriented Approach to Macintosh Application Development	46
• (H. Wertz) DVLSIP : An example of a Programming Environment	55
• (B. Seriat) Object Oriented Programming in Cedar	64
• (J.M. Hulot) SOS Interface, Un générateur d'Interfaces Homme-Machine	69
• (H. Lieberman) Delegation and Inheritance : Two Mechanisms for Sharing Knowledge in Object-Oriented Systems	79
• (J.P. Briot, P. Cointe, E. Saint-James) Réécriture et Récursion dans une Fermeture ; Etude sur un LISP à liaison superficielle, Application aux objets	90
• (P. Jeanjean) Analyse syntaxique de Smalltalk par redéfinition dynamique de fonctions virtuelles	101
• (O. Denry, O. Julien) Langage d'implémentation pour Langages Applicatifs Contribution à l'étude d'une relation informatique	114
• (A. Yonemura, H. Matsuda, E. Shoyama) An Approach to Object Oriented Concurrent Programming A Language AICS	125
• (P. Chussey) Comment programmer un multiprocesseur synchrone : connexion des données et propagation des signaux	135
• (W. Dang) Programmation Concurrente en Langage Orienté Objet	149
• (J.M. Geb) OMPHALE Système réparti orienté Objet	155
• (A. Anjewierden) How about a Prolog Object ?	167
• (Ph. Gautron) C++ : Instantiation, Héritage et Transmission de message	177
• (A. Collinot) GLISP et la Propagation de Contraintes	189
• (I. Borne) Langages orientés-objets et Education	198
• (D. Hollinger) Réseaux de Petri et Flavors pour la spécification et la simulation de systèmes productiques	206
• (E. Fimbel, F. Rousselot) PROMESS : Programmation par messages et descriptions fonctionnelles	216
• (E. Chouraqui, Ph. Dugerdit) Applications des Langages Orientés Objet à la CAO de l'architecture	226
• (D. Demigny, L. Kessal, T. Maurin, F. Devos) Les implications du traitement symbolique dans l'architecture des machines Exemple : le traitement de cas case of dans PCARL2	232

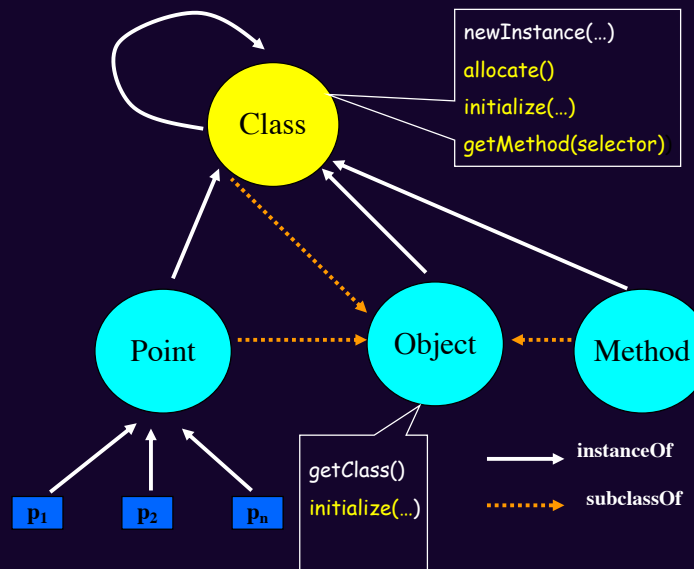
Bulletin publié avec le concours de l'AFCT-INFORMATIQUE



Contribution au modèle ObjVlisp

<https://en.wikipedia.org/wiki/ObjVlisp>

ObjVlisp a minimal model (IRCAM 1984)



Meta Object Protocols

Structural reflection

- **object creation**

- allocate o initialize**

- constructors
- metaclasses
- prototypes

- **class modification**

- adding/removing a field
- changing the class hierarchy

Behavioral reflection

- **message sending**

- lookup o apply**

- error handling
- class based inheritance
- prototype & delegation
- encapsulators
- wrappers & proxies

1986-1987. TIT (jpb) & CMI (pc)

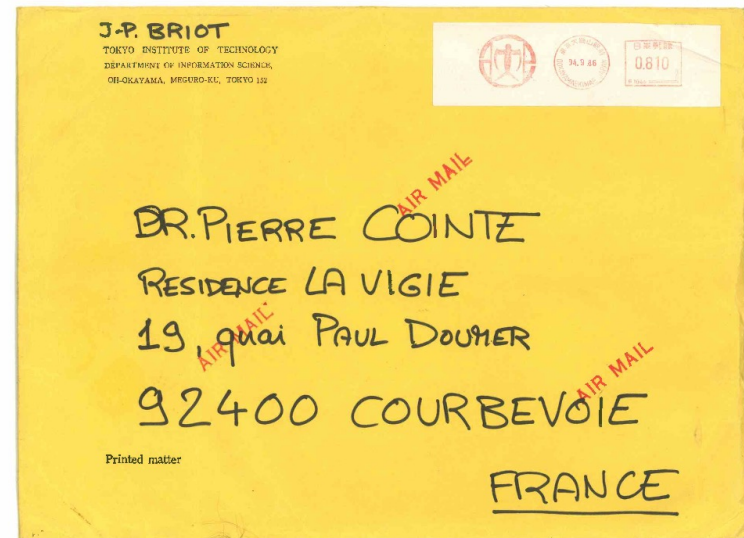
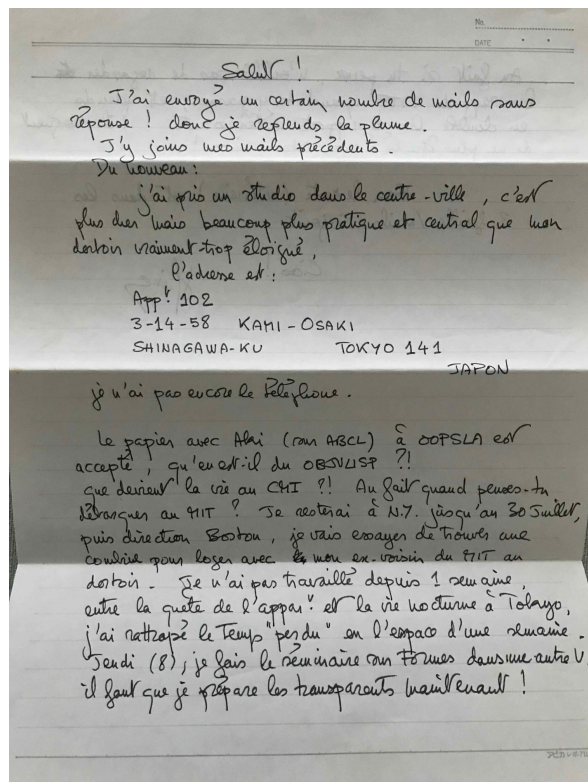
Portage du modèle ObjVlisp en Le Lisp, Pive, Scheme, Common Lisp ...



P. Gautron, B. Serpette, J.-J. Codani,

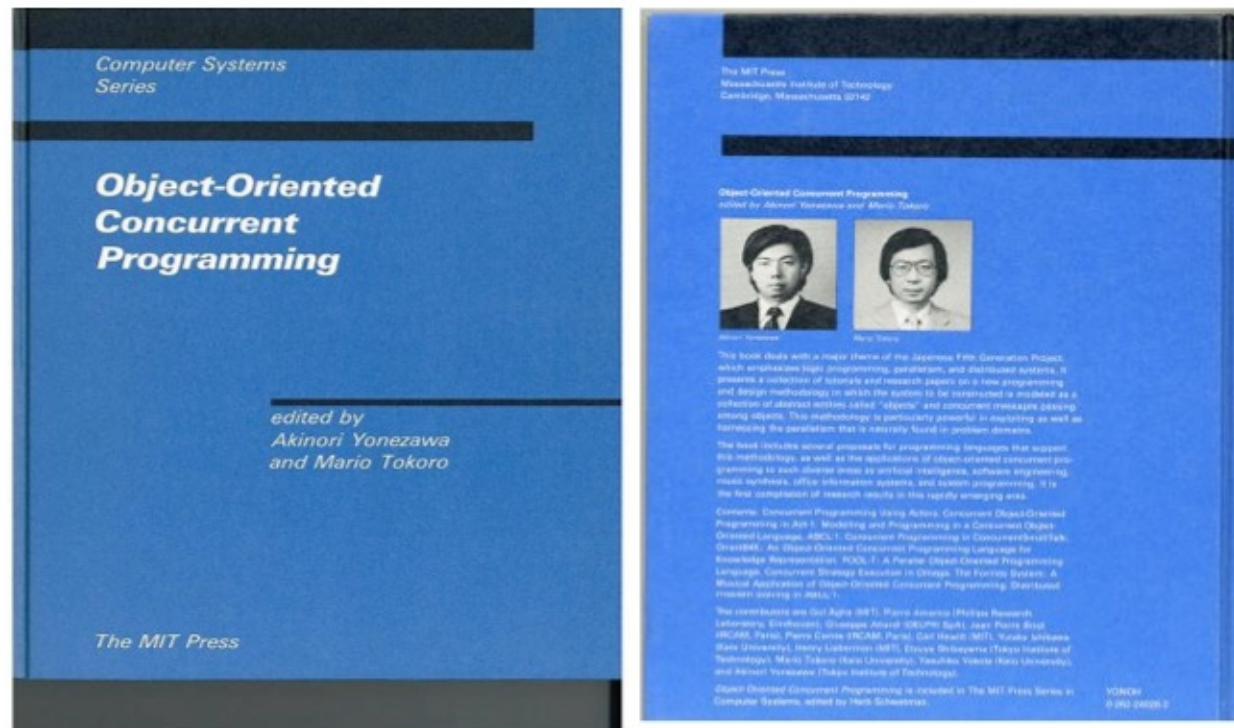
1986-1987. TIT (jpb) & CMI (pc)

Publications modèle ObjVlisp : ECAI 86, IJCAI 87 et OOSPLA 87

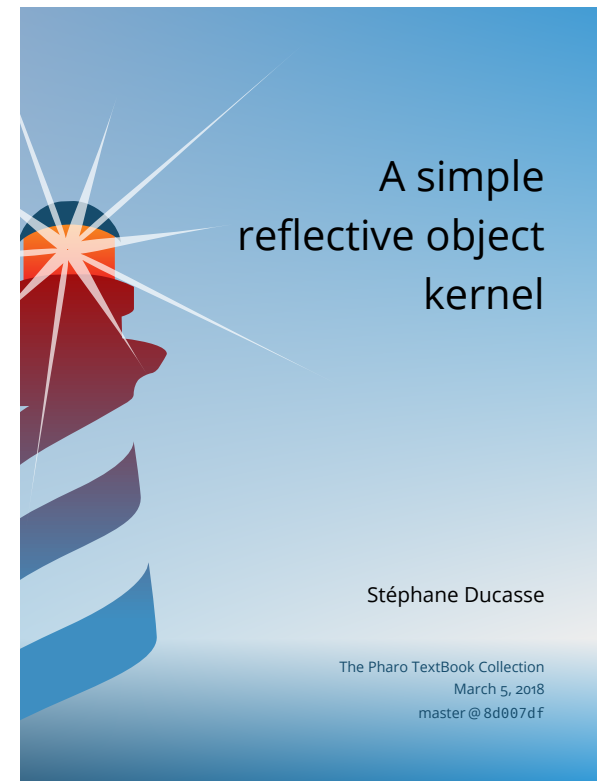


The MIT Press

The Formes System : A Musical Application of OOC

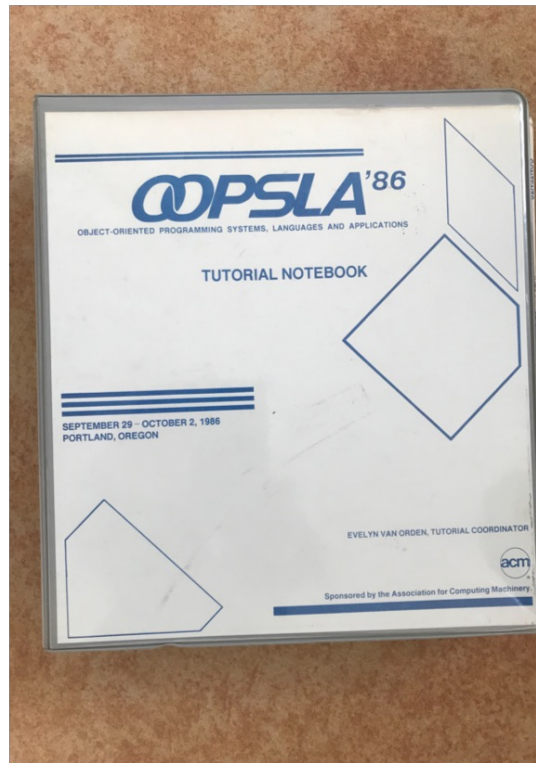


Vie du modèle ObjVlisp : 1987-2018



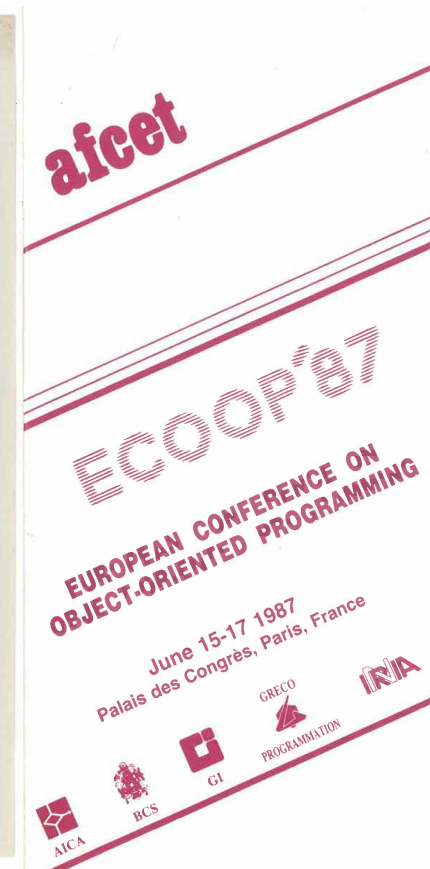
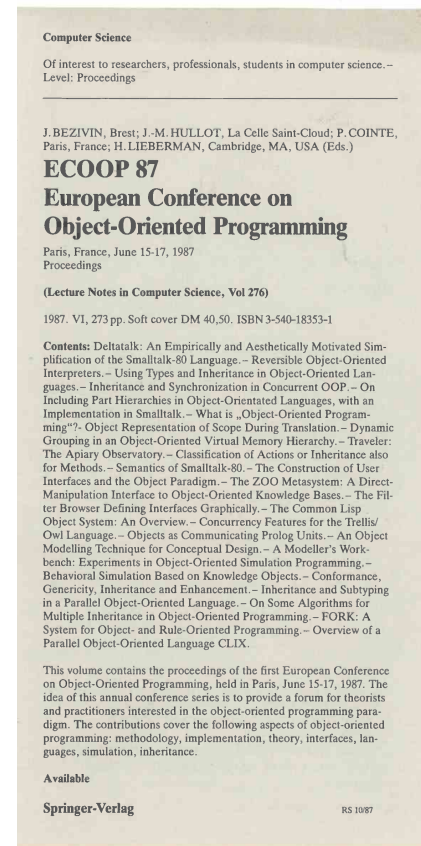
OOSPLA 86 @ Portland

OOCP in ABCL/1



ECOOP 87 @ Paris

Inheritance and Synchronisation in OOCP



L'ex-gadget de JJ-SS meurt d'une overdose de fric

LA dissolution pour « absence totale de résultats » du Centre mondial informatique prouve que les meilleures choses ont une fin. Créé en 1982 par Jean-Jacques Servan-Schreiber pour servir au rayonnement de la France (et accessoirement au sien), ce machin aura coûté à l'Etat environ 470 millions de francs. Soit, pour se faire une idée plus précise du pognon englouti : 47 milliards de centimes. On a fini par estimer, en haut lieu, que cela commençait à bien faire. Du coup, le 30 avril prochain, on va fermer la boutique.

Dès le début, « Le Canard » a raconté les frasques et les lubies coûteuses de JJ-SS. Mais, depuis le 1^{er} janvier 1986, sous la présidence de son successeur, Jean-Louis Funck-Brentano, les choses ne se sont guère améliorées. Le Centre a du plomb dans l'aile et ses 70 employés (dont 56 grosses têtes) errent dans les couloirs « sans projet ni boulot », disent-ils. Mais non sans fric. Le contrôleur financier — un pervers ! — s'est amusé à diviser le total des frais de fonctionnement par le nombre de personnes censées fonctionner. Et il a trouvé 410 000 F par tête de pipe pour l'année 1986, plus — toujours par tête de pipe — un salaire moyen de 200 000 F.

On arrête donc les frais et on vire tout le monde. Cette délicate épuration est confiée à Hervé de Charette, ministre de la Fonction publique et du Plan. Et là, l'histoire devient franchement cocasse.

Silence, on coule !

Comme les personnels débarqués n'ont pas le sentiment d'avoir vraiment démerité de la patrie, ils mettent des bâtons dans les roues de Charette et refusent en bloc les deux plans sociaux que ce ministre leur propose. Les deux formules de licen-

ciement sont pourtant du libéralisme pur jus. Il y a le plan A et le plan B, assortis d'une « note d'accompagnement » qui incite vivement à signer la formule B.

Y aurait-il un truc ? En échange d'une série d'engagements hors plan social, très avantageux financièrement pour eux, les virés sont priés de se désintéresser du sort du matériel informatique, lequel sera « racheté ou dévolu » à diverses administrations ou entreprises. En clair : le gouvernement casque et on doit lui faire confiance.

Ordinateurs qui volent

Le Centre mondial, c'est une véritable caverne d'Ali Baba informatique. L'inventaire du matériel qui a fini par être établi est à tomber à la renverse. Et par deux fois. La première fois, à cause des trésors qu'il met au jour ; la seconde, à cause de ceux qu'il oublie. Curieusement, le matériel le plus précieux passe à

l'as. Les personnages qui ont dressé l'inventaire n'ont pas oublié un seul Minitel. Mais, disent certains cadres du Centre, ils sont passés sans les voir à côté de plusieurs engins haut de gamme : un Vax 785, un Vax 8600 et un Dec 2060.

Les ignorants ne broncheront pas, mais les initiés apprécieront : le prix total de ces trois seules bécasses frôle les 10 millions de francs. De plus, selon les mêmes grinchoux, trois imprimantes laser et quelques autres babioles très performantes n'ont pas été signalées dans l'inventaire. Elles devaient être montées sur roulettes.

André Rougeot

1987

Fermeture du CMIRH en avril.

Création équipe mixte RXF-LITP Retour JPB @Paris CR CNRS

Coopération avec l'université

C'est au vingt-sixième étage de la tour centrale de l'Université Pierre et Marie Curie (Jussieu) que s'est déroulée, le 2 avril dernier, l'inauguration de l'accord de coopération récemment signé entre Rank Xerox France, les universités Paris VI et VII et le CNRS.

Cet accord est la formalisation de travaux de recherche engagés depuis deux ans et conduits par une équipe mixte, composée de trois chercheurs de l'entreprise et de quatre chercheurs provenant soit du CNRS, soit du monde

universitaire. Les engagements réciproques des contractants sont forts et exemplaires à plus d'un titre. Forts dans la mesure où ceux-ci relèvent d'objectifs communs de recherche à atteindre dans le domaine de la programmation par objets et pour lesquels chaque partenaire apporte des moyens matériels et humains. Exemplaires parce qu'ils sont la concrétisation de rapprochements entre le monde de la recherche publique et le monde industriel qui en France sont encore exceptionnels dans leur mise en œuvre. Depuis deux ans maintenant, l'équipe mixte RXF-LITP du nom de l'entreprise et du laboratoire associé (Laboratoire informatique théorique et programmation), travaille, certes discrètement, mais avec efficacité, et quelques succès sont déjà à mettre à son actif :

— la spécification de nouveaux



De gauche à droite : Bernard Reibell, Directeur Général Adjoint de Rank Xerox, Jacques Sakarovich, Directeur de l'Institut Blaise Pascal, Dominique Perrin, Directeur du LITP, Michel Garnier, Président de l'Université Paris VI.

mécanismes « objet » et « acteur », — un financement de la Communauté européenne pour des recherches sur une machine multiprocesseur, — l'organisation du prochain congrès international sur la programmation par objets ECOOP-OOPSLA'90 au Canada.

David Kearns : «Assigner Apple en justice était notre seul recours»

Le Président du Groupe Xerox, David Kearns, a expliqué que Xerox n'avait aucun autre recours que d'assigner Apple Computer Inc. devant les tribunaux, pour utilisation illégale d'un logiciel

couvert par un copyright. Xerox accuse Apple de proposer pour Lisa et Macintosh des logiciels dérivant directement de travaux originaux conduits par des chercheurs de Xerox. «Xerox n'est pas une compagnie procédurière, a souligné David Kearns, mais notre seul recours était de demander réparation auprès des tribunaux pour des agissements que nous estimons illégaux et déloyaux, et qui, s'ils continuent à être perpétrés impunément, vont à l'encontre des intérêts de Xerox, de ses actionnaires et du secteur professionnel de l'informatique.» David Kearns a ajouté que toutes les tentatives de Xerox visant à une conciliation amiable — qui comprenaient une proposition de concession à Apple d'une licence pour le logiciel concerné — se sont heurtées à un refus.

1987-1991. Equipe RXF-LITP tour 55-65



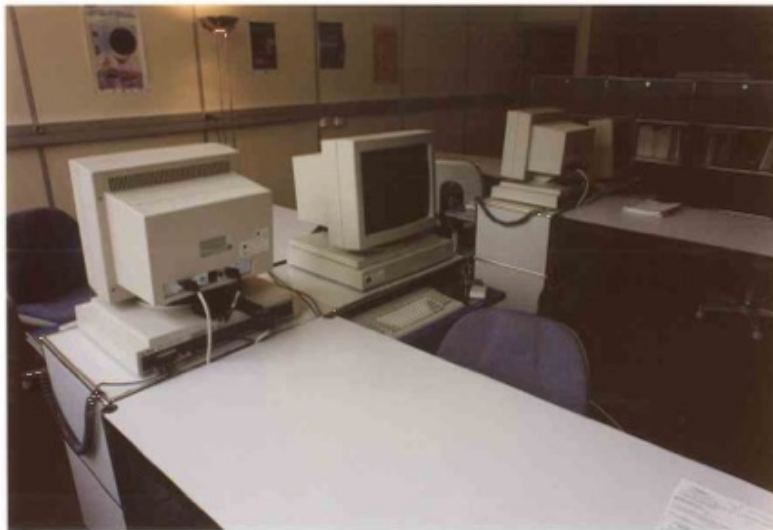
J. Sakarovitch, C. Dony, pc, jpb, P. Gautron, G. Dupoirier
et Françoise Boquillon



Jpb et C. Dony



Machines SUN (GRECO)
TNODE (ESPRIT)
Meubles USM



1988. DESS GLA Méribel

Responsable : Max FONTET, Paris 6 - LITP

L. Bougé, CNRS
J.-P. Briot, LITP
P. Gointe, LITP
P.-L. Curien, CNRS
J. Ferber, Paris 6 - LAFORIA
C. Gomez, INRIA
D. Lazard, Paris 6 - LITP
R. Lescœur, BULL
J.-F. Perrot, Paris 6 - LAFORIA
C. Roucairol, Paris 6 - MASI
F.-X. Testard-Vaillant, LITP
R. Voyer, Paris 6 - LAFORIA



DESS - GÉNIE DES LOGICIELS APPLICATIFS

organisation

Temps plein ou 1^{er} extern au 30 avril suivi d'un **stage en entreprise** de 3 mois entre le 1^{er} mai et le 30 septembre.

capacité d'accueil

Ce DESS est prévu pour accueillir 2^e étudiants. Parmi ces étudiants, certains peuvent suivre cet enseignement au titre de la formation permanente.

renseignements pédagogiques et techniques

Bernard Aïme
Laboratoire d'Informatique Théorique et Programmation
Tour 05, Rotonde, 1^{er} étage
Tél : 43 00 25 25, poste 3084

renseignements administratifs et inscriptions

- Au titre de l'enseignement normal :
Scolarité
Aile 55-65 - 1^{er} étage - porte 01
Lundi, mardi, jeudi 9 h 30 - 11 h 30 et 13 h 30 - 16 h 30
Vendredi, vendredi 9 h 30 - 11 h 30.
- Au titre de la formation permanente :
Formation permanente
Tour Centrale - 1^{er} étage - porte 07
Lundi, mardi, vendredi 9 h 12 - 11 h 12 et 13 h 30 - 16 h 30
Mardi 10 h - 16 h, vendredi 10 h - 14 h.

Université Pierre et Marie Curie (Paris 6)
Laboratoire Informatique Théorique et Programmation
4, place Jussieu - 75252 Paris Cedex 05



UNIVERSITÉ PIERRE ET MARIE CURIE (PARIS 6)
LABORATOIRE INFORMATIQUE THÉORIQUE ET PROGRAMMATION
(UA au CNRS n° 246)



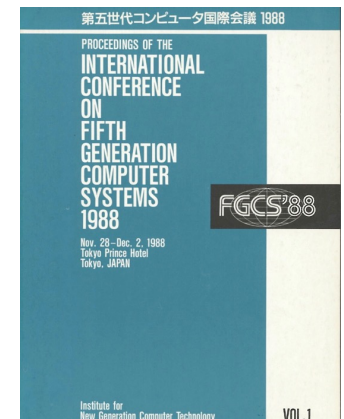
UNIX*
DESS
GÉNIE DES LOGICIELS
APPLICATIFS

* Marque déposée de l'Université UCL.

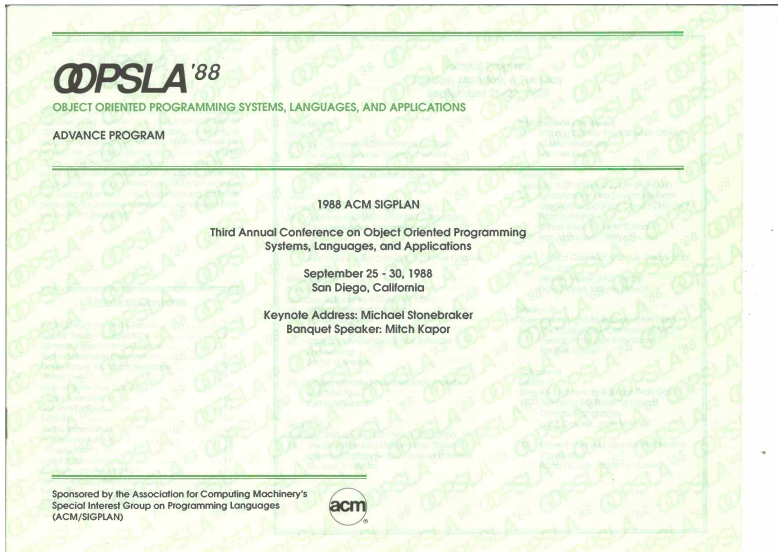
Visite du Tokyo Sony Lab pendant FGCS'88



1. Design of a concurrent language for distributed artificial intelligence. Briot & Ferber
2. A tutorial introduction to metaclass architectures as provided by class oriented languages. Cointe

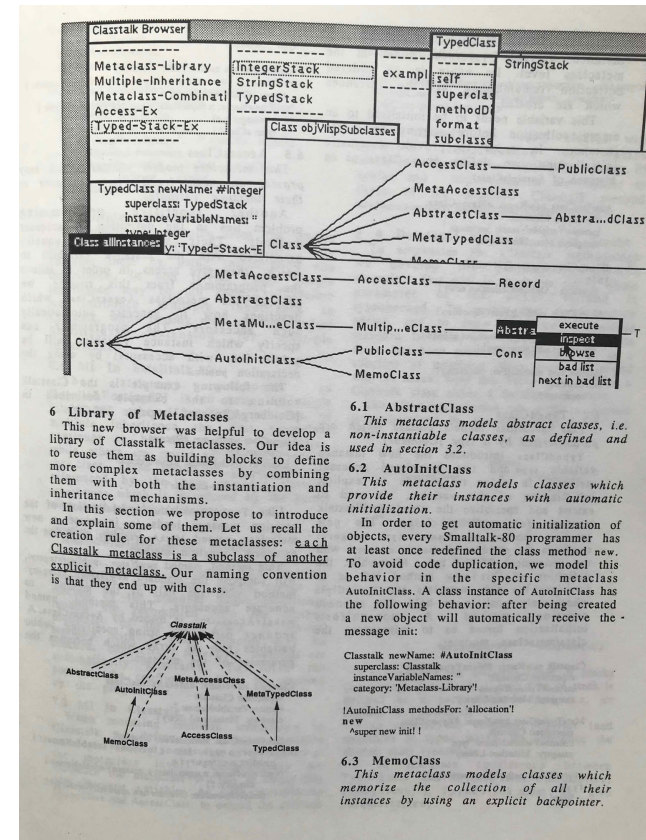


OOPSLA 88 @ San Diego

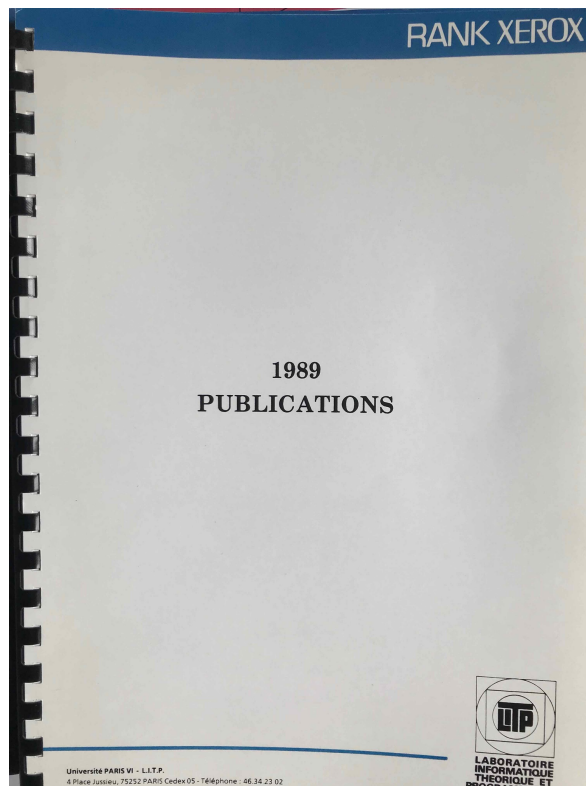


OOPSLA 89 @ New Orleans

Programming with explicit metaclasses in
Smalltalk-80



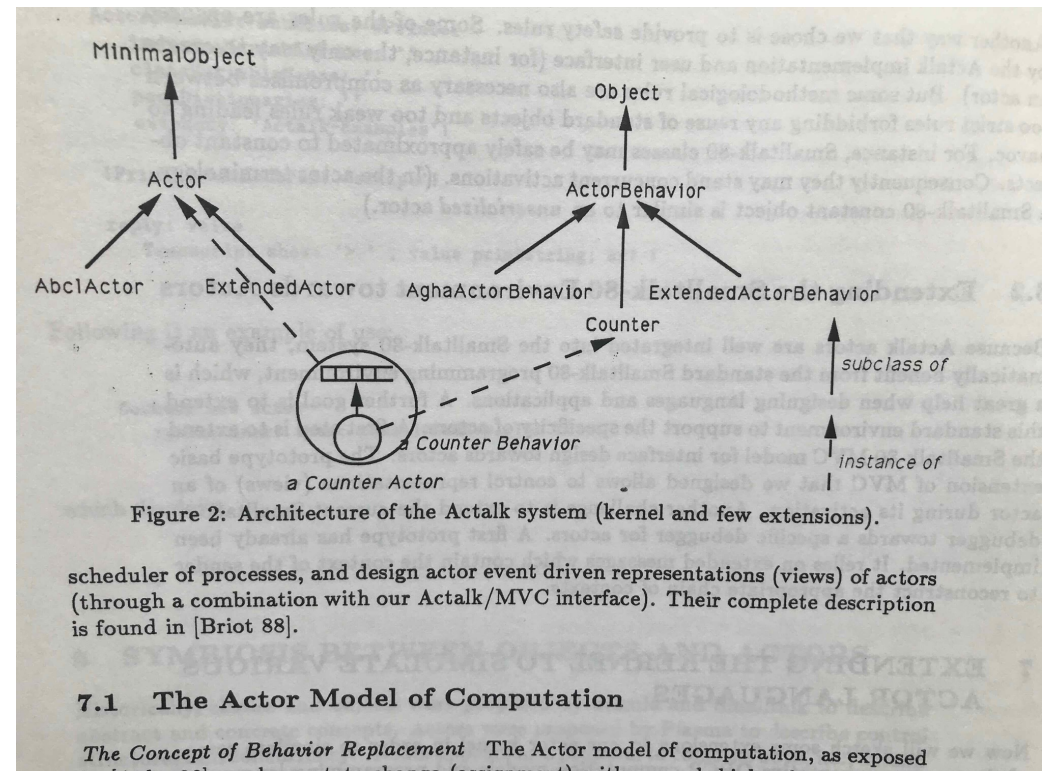
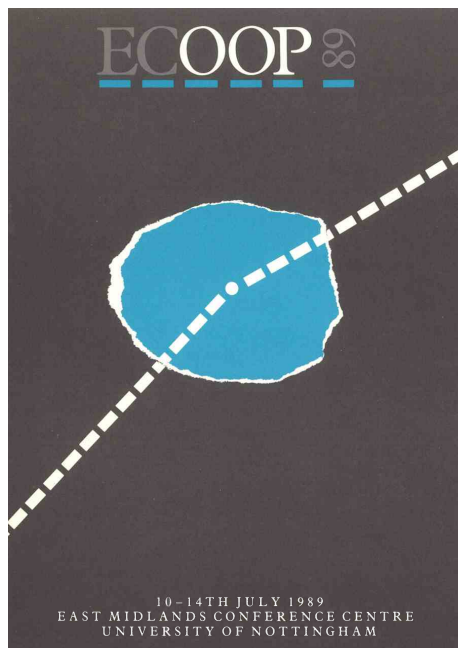
Publications à ECOOP, OOPSLA voire TOOLS après rodage de l'écriture à RFIA et GLA au niveau national



Sommaire / Contents	
Avant-Propos Jacques Sakarovich & Gérard Dupoiner	
Préface / Foreword Pierre Cointe	
[1] Actalk : A Testbed for Classifying and Designing Actor Languages in the Smalltalk-80 Environment Jean-Pierre Briot Proceeding of the third European Conference on Object-Oriented Programming (ECOOP'89), Cambridge University Press - Editor Steve Cook - Pages 109 - 129 Nottingham, Juillet 1989	
[2] Persistence and Migration for C++ Objects Marc Shapiro, Philippe Gautron & Laurence Mosseri Proceeding of the third European Conference on Object-Oriented Programming (ECOOP'89), Cambridge University Press - Editor Steve Cook - Pages 191 - 204 Nottingham, Juillet 1989	
[3] OOCPP = OOP + C Jean-Pierre Briot Position statements for the ECOOP'89, Workshop on Object Based Concurrency Nottingham, Juillet 1989	
[4] Programming with Explicit Metaclasses in Smalltalk-80 Jean-Pierre Briot & Pierre Cointe Proceeding of OOPSLA'89 - Editor N. Meyrowitz - Pages 419 - 431 New Orleans (Louisiane), Octobre 1989	
[5] Metaclass Compatibility Nicolas Graube Proceeding of OOPSLA'89 - Editor N. Meyrowitz - Pages 305 - 315 New Orleans (Louisiane), Octobre 1989	
[6] An introduction to the C++ task system Philippe Gautron The C++ Report Journal - Volume 1, Number 9 Octobre 1989	
[7] Actalk : une Plateforme de Modélisation de Langages d'Acteurs en Smalltalk 80 Jean-Pierre Briot Proceeding of the seventh "Reconnaissance des Formes et Intelligence Artificielle" Conference (RFIA'89), Tome 1 - Pages 147 - 162 Paris, Novembre 1989	
[8] ClassTalk : Une transposition des métaclasses ObjVilisp à Smalltalk-80 Pierre Cointe & Jean-Pierre Briot Proceeding of the seventh "Reconnaissance des Formes et Intelligence Artificielle" Conference (RFIA'89), Tome 1 - Pages 127 - 146 Paris, Novembre 1989	
[9] Apports du formalisme objet aux problèmes de la gestion des exceptions Christophe Dony Deuxièmes journées internationales sur le Génie Logiciel et ses applications, Toulouse, Décembre 1989	

10/07/1989. Soutenance HDR Des objets aux acteurs : 7 ans de réflexion

J. Bézivin, PC, P. Greussay, H. Lieberman, JFP, P. Sallé et A. Yonezawa



Actalk: a testbed for classifying and designing actor languages in the Smalltalk-80 environment

ECOOP 87@Paris



ECOOP-OOSPLA 90@ Ottawa



1981-1991. Ils étaient doctorants

- Bernard Serpette, Isabelle Borne, François Testard Vaillant et Patrice Boizumault
- Philippe Gautron
- Jacques Ferber

- Alain Deutsch et Charles Consel
- Jean-Jacques Codani et Nicolas Graubé
- Loïc Lescaudron et Sylvie Lemarié (TACT)
- David Plainfossé et Hayssam Saleh (TNODE Transputers)
- Jacques Malenfant (Montréal)

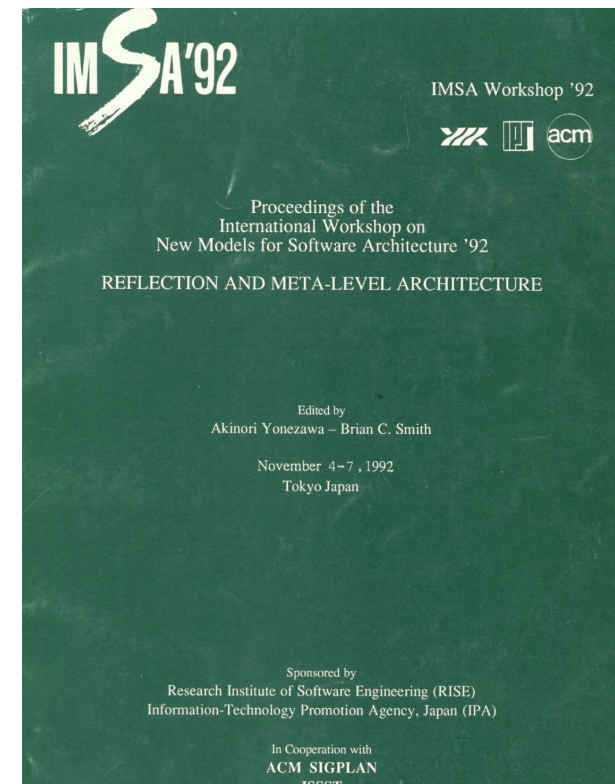
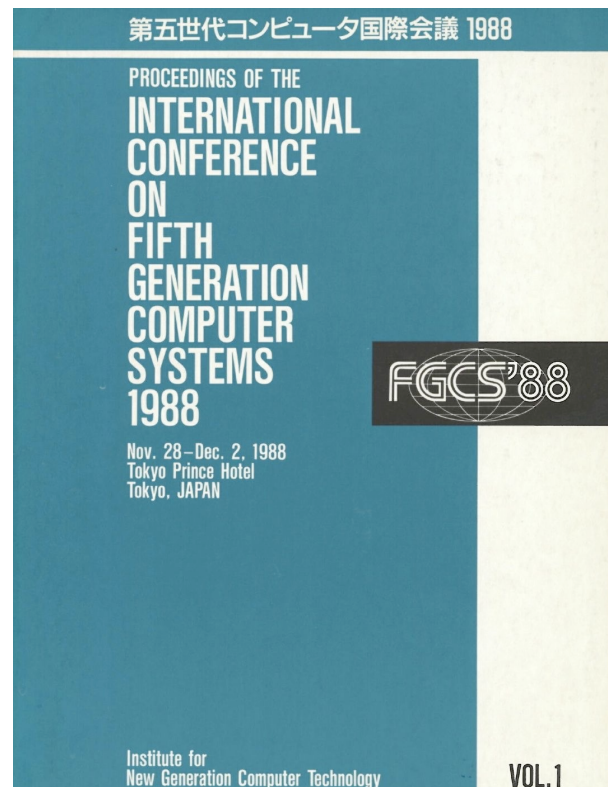
- Rachid Guerraoui (CEA, Mines Paris)

- Olivier Danvy, Francis Wolinski et François Pachet

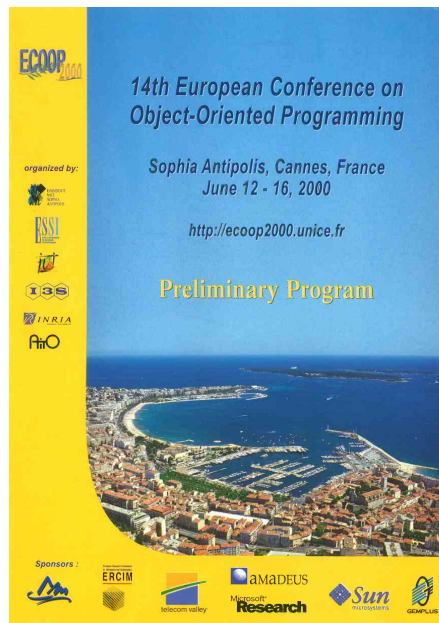
1981-1991. Les rencontres avec

- Xavier Rodet
 - Patrick Greussay, Harald Wertz et Daniel Goossens
 - Jérôme Chailloux, Jean-Marie Hullot et Gérard Huet
 - Emmanuel Saint James et Eugen Neidl
 - Christian Queinnec
 - Jean-Louis Durieux et Patrick Salle
-
- Akinori Yonezawa et Mario Tokoro
 - Lee Gasser et Gul Agha
 - Henry Lieberman et Carl Hewitt
 - Peter Deutsch et Adèle Goldberg
 - Luc Steels et Pattie Maes
 - Danny Bobrow, Jim des Rivières, Brian Cantwell Smith et Gregor Kiczales

Retour à Tokyo de FGCS'88 à IMSA'92



ECOOP2000@Nice : le temps des retrouvailles



De gauche à droite :

S. Matsuoka, G. Kiczales, PC, A. Yonezawa, M. Aksit, J. Bezivin et JPB