

Comprendre la programmation par aspects

Pierre Cointe

*Projet Obasco (EMN - INRIA)
Ecole des Mines de Nantes
4, rue Alfred Kastler, BP 20722, F-44307 Nantes cedex 3
cointe@emn.fr & www.irisa.fr/obasco/*

RÉSUMÉ. La programmation par aspects s'attaque au problème général de la séparation des préoccupations en proposant un ensemble de techniques permettant d'isoler des unités de programme transverses (par exemple des services comme l'affichage, la persistance, le transport) pour adapter les applications (décrites par exemple par des classes) à ces unités. Dans cet exposé, nous présenterons d'abord certaines des intuitions à la base de l'approche par aspects. Nous décrirons ensuite certaines des limites de la programmation par objets, en particulier en terme de réutilisation, limites qui entraînent dispersion et mélange de certains codes. Nous introduirons alors quelques exemples choisis de programmation par aspects écrits principalement avec le langage AspectJ vu comme une extension du langage Java. Ces exemples seront l'occasion de caractériser le modèle d'aspects sous-jacent en précisant les notions de points de jonction, de coupes, d'introductions mais également de préciser le processus de tissage. Nous présenterons finalement une sélection des principaux travaux en cours dans la communauté AOSD. Nous conclurons en inscrivant l'approche par aspects dans le cadre plus générale de la programmation dite générative.

ABSTRACT. Aspect-Oriented programming (AOP) is another attempt to deal with separation of concerns. AOP provides techniques to represent crosscutting program units (for example services such as display, persistency, transport) and to weave these units with standard application (built for example with class hierarchies). In this talk, we will sketch some intuitions behind the AOP approach. We will discuss some limitations of the OOP approach focusing on code reusability and scalability (management of scattered and tangled code). We will present some examples of AOP written in AspectJ. Those examples will be the opportunity to describe the underlying model by introducing the notions of pointcut, advice, introduction, cflow and weaving. Then we will present the main research tracks open by the AOSD community. This will be the opportunity to consider the AOP approach in the more general context of generative programming.

MOTS-CLÉS : séparation des préoccupations, réutilisation, tissage, objet, classe, points de jonction, coupe, aspect, www.eclipse.org/aspectj, langage dédié, programmation générative.

KEYWORDS: separation of concerns, reusability, weaving, object, class, pointcut, advice, aspect, www.eclipse.org/aspectj, domain specific language, generative programming.

Présentation à CARI 2004, le 25 novembre 2004.



1. Appel de bibliographie en fonction du domaine

- 1) Objets [AKS 96], [LIE 96], [GAB] et [COI 04b].
- 2) Réflexion [SMI 82], [PAE 93], [COI 99], [THO 02] et [TAN 03].
- 3) Aspects [KIC 97], [KIS 03], [DOU 01], [WAN 03], [COI 04a], [ASP], [LIE 04] et [AKS 04].
- 4) Composants [CZA 00]et [GPC].

2. Bibliographie

- [AKS 96] AKSIT M., BLACK A., CARDELLI L., COINTE P., GUERRAOUR R., AL, « Strategic Research Directions in Object Oriented Programming », *ACM Computing Surveys*, vol. 28, n° 4, 1996, p. 691-700.
- [AKS 04] AKSIT M., CLARKE S., ELRAD T., FILMAN R. E., Eds., *Aspect-Oriented Software Development (à paraître)*, Addison-Wesley, 2004.
- [ASP] ASPECTJ, « AspectJ sites », see <http://eclipse.org/aspectj> but also <http://www.parc.com/research/csl/projects/aspectj/downloads/>.
- [COI 99] COINTE P., Ed., *ACM Meta-Level Architectures and Reflection, Second International Conference, Reflection 99*, vol. 1616 de *Lecture Notes in Computer Science*, Saint-Malo, France, juillet 1999, Springer-Verlag.
- [COI 04a] COINTE P., Ed., *Première Journée Francophone sur le Développement de Logiciels Par Aspects*, Ircam, Paris, septembre 2004, on line proceedings available at : <http://www.emn.fr/x-info/obasco/events/jfdlpa04>.
- [COI 04b] COINTE P., NOYÉ J., DOUENCE R., LEDOUX T., MENAUD J.-M., MULLER G., SÜDHOLT M., « Programmation post-objets : des langages d aspects aux langages de composants », *RSTI L Objet, colloque en l honneur de Jean-François Perrot (à paraître)*, vol. 10, n° 4, 2004, Lavoisier, see also : <http://www.lip6.fr/colloque-JFP/>.
- [CZA 00] CZARNECKI K., EISENECKER U. W., *Generative Programming. Methods, Tools and Applications*, Addison Wesley, 2 rd printing édition, 2000.
- [DOU 01] DOUENCE R., MOTELET O., SÜDHOLT M., « A formal definition of crosscuts », YONEZAWA A., MATSUOKA S., Eds., *Proceedings of the 3rd International Conference on Reflection 2001*, vol. 2192 de *Lecture Notes in Computer Science*, Kyoto, Japan, septembre 2001, Springer-Verlag, p. 170–186.
- [GAB] GABRIEL R., « Resolved: Objects Have Failed », see <http://www.dreamsongs.com/Essays.html> and also <http://www.lip6.fr/colloque-JFP/>.
- [GPC] GPCE, « Home page of the international conference on Generic Programming and Component Engineering », see <http://www.cs.rice.edu/~faha/gpce/>.
- [KIC 97] KICZALES G., LAMPING J., MENDHEKAR A., MAEDA C., LOPES C., LOINGTIER J.-M., IRWIN J., « Aspect-Oriented Programming », AKSIT M., MATSUOKA S., Eds., *ECOOP 97 - Object-Oriented Programming - 11th European Conference*, vol. 1241 de *Lecture Notes in Computer Science*, Jyväskylä, Finland, juin 1997, Springer-Verlag, p. 220–242.
- [KIS 03] KISELEV I., *Aspect-Oriented Programming with AspectJ*, Sams Publishing, 2003.





- [LIE 96] LIEBERHERR K. J., *Adaptive Object-Oriented Software: The Demeter Method with Propagation Patterns*, PWS Publishing Company, Boston, 1996, ISBN 0-534-94602-X.
- [LIE 04] LIEBERHERR K., Ed., *Proceedings of the 3rd international conference on Aspect-Oriented Software Development*, Lancaster, UK, mars 2004, ACM Press, see also : <http://aosd.net>.
- [PAE 93] PAEPCKE A., Ed., *Object-Oriented Programming: The CLOS Perspective*, The MIT Press, 1993.
- [SMI 82] SMITH B., « Procedural Reflection in Programming Languages », PhD thesis, Massachusetts Institute of Technology, 1982.
- [TAN 03] ÉRIC TANTER, NOYÉ J., CAROMEL D., COINTE P., « Partial Behavioral Reflection: Spatial and Temporal Selection of Reification », CROCKER R., STEELE JR. G. L., Eds., *Proceedings of the 18th ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications (OOPSLA 2003)*, ACM Press, October 2003, p. 27-46.
- [THO 02] THOMAS D., « Reflective Software Engineering - From MOPS to AOSD », *Journal Of Object Technology*, vol. 1, n° 4, 2002, p. 17-26.
- [WAN 03] WAND M., « Understanding Aspects », 2003, Invited talk at ICFP 2003, see <http://www.ccs.neu.edu/home/wand/>.

