

Adaptation fonctionnelle de composants gros-grain avec JBOSS/AOP

O.Caron¹, B.Carré¹, A.Muller¹, G.Vanwormhoudt^{1,2}

¹ Laboratoire d'Informatique Fondamentale de Lille
UMR CNRS 8022, Université des Sciences et Technologies de Lille

² TELECOM Lille I

Cité scientifique, 59655 Villeneuve d'Ascq cedex

Les technologies à composants constituent une avancée certaine dans la quête de réutilisation, par leur potentiel de composition et d'adaptation de leurs propriétés fonctionnelles ou non-fonctionnelles. Nous nous intéressons ici plus particulièrement aux composants de systèmes d'information (SI) tels qu'offerts par la technologie EJB et à leur structuration fonctionnelle aux travers de vues. Dans ce domaine, les applications sont structurées en composants "gros-grain" [8, 11], constitués d'objets ou de composants plus élémentaires qui concourent à réaliser une fonction spécifique du SI.

Aux niveaux modélisation et conception, de nombreuses méthodologies aident à décomposer de tels systèmes relativement à leurs dimensions fonctionnelles telles que Catalysis [8], Subject-Oriented Design [7], Role-Oriented Design [15], View-Oriented Design [5]. Plus récemment, conformément aux approches MDA (Model-Driven Architecture) puis MDE (Model Driven Engineering) [12], ces méthodologies ont concrétisé leurs produits sous la forme de modèles. Dans la lancée, leurs capacités de réutilisation et d'adaptation ont été augmentées par recours à des techniques de paramétrage ("templates") [3, 6, 1].

Nous prenons comme point de départ nos travaux sur les modèles paramétrés [13] et les chaînes de production logicielle qu'ils autorisent selon des modes de ciblage et des technologies différentes [14]. Nous présentons ici plus particulièrement une voie qui consiste à produire une architecture à base de composants vues gros-grain dans la technologie EJB. Pour être fonctionnellement réutilisables, ces composants doivent être indépendants de tout domaine d'application et donc adaptables dans leurs propriétés fonctionnelles. Leur application à un système particulier nécessite alors un mécanisme de connexion que nous présentons.

Pour cela, l'architecture retenue est fondée sur notre patron de conception de vues [2] couplé au patron Adaptateur [10, 4]. La mise en œuvre profite des capacités des conteneurs d'applications extensibles [9] qui exploitent l'AOP pour injecter de nouveaux services. De telles capacités sont offertes par le serveur d'applications JBOSS que nous utilisons pour décrire les connexions à l'aide d'annotations et implanter le mécanisme de connexion par interception dynamique. Les composants EJB vues résultats sont entièrement compatibles et "connectables" à une architecture de composants EJB standard.

References

1. E. Baniassad and S. Clarke. Theme: An approach for aspect-oriented analysis and design. In *ICSE '04: Proceedings of the 26th International Conference on Software Engineering*, pages 158–167. IEEE Computer Society, 2004.
2. O. Caron, B. Carré, A. Muller, and G. Vanwormhoudt. A Framework for Supporting Views in Component Oriented Information Systems. In *Proceedings of International Conference on Object Oriented Information Systems (OOIS'03)*, volume 2817 of *LNCS*, pages 164–178. Springer, September 2003.
3. O. Caron, B. Carré, A. Muller, and G. Vanwormhoudt. An OCL Formulation of UML 2 Template Binding. In *Proceedings of 7th International Conference on The Unified Modeling Language. Model Languages and Applications (UML 2004)*, volume 3273 of *LNCS*, pages 27–40. Springer, October 2004.
4. O. Caron, B. Carré, A. Muller, and G. Vanwormhoudt. Mise en oeuvre d'aspects fonctionnels réutilisables par adaptation. *Revue L'objet, Programmation par Aspects, Hermes Ed.*, 11(3):105–118, January 2005.
5. Olivier Caron, Bernard Carré, and Laurent Debrauwer. Contextualization of OODB schemas in CROME. In *Database and Expert Systems Applications*, pages 135–149, 2000.
6. T Clark, A Evans, and S Kent. A Metamodel for Package Extension with Renaming. In *The Unified Modeling Language 5th International Conference*, Proceedings LNCS 2460, pages 305–320, Dresden, Germany, September 2002.
7. S. Clarke, W. Harrison, H. Ossher, and P. Tarr. Subject-Oriented Design: Towards Improved Alignment of Requirements, Design and Code. In *Object-Oriented Programming Systems, Languages and Applications (OOPSLA)*, Denver, 1999.
8. Desmond D'Souza and Alan Wills. *Objects, Components and Frameworks With UML: The Catalysis Approach*. Addison-Wesley, 1999.
9. M. Fleury and F. Reverbel. The JBoss extensible server. In M. Endler and D. Schmidt, editors, *Middleware 2003 — ACM/IFIP/USENIX International Middleware Conference*, volume 2672 of *LNCS*, pages 344–373. Springer-Verlag, 2003.
10. E. Gamma, R. Helm, R. Johnson, J. Vlissides, and G. Booch. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional Computing, USA, 1995.
11. David Helton. Coarse-Grained Components as an Alternative to Component Frameworks. In *ECOOP Workshops*, page 188, 1999.
12. S Kent. Model Driven Engineering. In *Proceedings of IFM 2002*, LNCS 2335, pages 286–298. Springer-Verlag, 2002.
13. A. Muller, O. Caron, B. Carré, and G. Vanwormhoudt. On Some Properties of Parameterized Model Application. In *First European Conference on Model Driven Architecture - Foundations and Applications (ECMDA-FA'05)*, volume 3748 of *LNCS*, pages 130–144. Springer, November 2005.
14. Alexis Muller, Olivier Caron, Bernard Carré, Gilles Vanwormhoudt, and Salim Bouzitouna. Ingénierie multi-modèles : Projection flexible d'assemblages de modèles. In *Conf. francophone Langages et Modèles à Objets (LMO'07)*, Toulouse, mars 2007.
15. T. Reenskaug, P. Wold, and O.A. Lehne. *Working with Objects: The OORam Software Engineering Method*. Prentice-Hall, Inc., 1995.