
Vers un environnement d'assemblage de composants logiciels

Nicolas Belloir

*Laboratoire d'informatique LIUPPA, Université de Pau,
BP 1155, 64013 Pau CEDEX
nicolas.belloir@univ-pau.fr*

1. L'assemblage de composants : une problématique encore ouverte

L'ingénierie logicielle basée composant a pour objectif d'améliorer le développement d'applications par assemblage de *composants logiciels*. Dans ce contexte, la construction d'applications présente de multiples avantages notamment en terme de réutilisation de code, de qualité de développement, de modularité, etc. Les nombreux travaux menés sur ce thème ont permis des avancées importantes et la maturité des architectures distribuées du type CORBA a d'ailleurs contribué à ce développement en fournissant des modèles canoniques de composants. Néanmoins, les méthodes de développement et même les notations normalisées comme UML n'ont pas suivi cette montée en puissance. Cette dernière présente des incohérences notables dans les concepts qu'elle propose pour traiter la composition logicielle (Henderson-Sellers *et al.*, 1999), point névralgique de l'assemblage de composants. De plus, la mise en œuvre de la composition logiciel lors de la phase d'implémentation est souvent difficile. L'assemblage de composants logiciels reste donc un problème complexe et délicat pour les concepteurs. Pour remédier à cela, il est nécessaire de prendre en compte la composition logiciel au plus tôt dans le cycle de vie d'un logiciel. Dans ce cadre, nous cherchons à étudier et à développer un environnement cohérent (méthode, outils, librairie, ...) supportant le développement d'applications à base de composants et ceci, de la phase d'analyse à la phase de test.

2. Une démarche en deux axes complémentaires

Nous travaillons de façon complémentaire sur deux aspects s'intégrant dans un cadre cohérent et pratique pour cette thèse. Le premier aspect porte sur une étude fondamentale de la composition/combinaison, au travers, notamment, de l'étude exhaustive du concept plus général de relation Tout-Partie (Belloir *et al.*, 2001b). Cette dernière est une généralisation de la notion d'agrégation/composition. Via ce

type de relation, une méthode de combinaison de composants logiciels doit être définie. Pour ce faire, nous avons défini un ensemble de propriétés spécifiques à ce type de relation. A partir de là, nous allons en mener une étude formelle, notamment au niveau de leurs différentes interactions, afin de caractériser formellement les différents types de relations Tout-Parties. Ces derniers serviront de modèles dans la conception d'applications basées composants et en guideront le développement. Le deuxième aspect auquel nous nous intéressons, porte sur l'évaluation et la certification de la composabilité d'un composant avec son environnement. En effet, il faut s'assurer que le composant à réutiliser fournit bien les services qu'il est censé fournir et qu'il a bien le comportement qu'il est censé avoir. Nous avons, dans le cadre du projet européen *Component+* (<http://www.component-plus.org/>), développé une technique permettant l'intégration directe des protocoles de test dans les composants (Built-In Test: BIT). Le principe consiste à fournir non plus simplement le composant initial, mais un composant BIT fournissant des interfaces génériques de test. Le client de ce composant peut donc, grâce à ces interfaces, manipuler le composant pour le tester. Nous avons développé une librairie java, basée sur le principe de la réflexion, permettant de mettre en œuvre cette technique (Belloir *et al.*, 2001a). Cette librairie offre la possibilité de décrire le comportement du composant à l'aide de la notion de statecharts, puis de manipuler le composant BIT et ses interfaces. De plus, elle permet à l'utilisateur du composant de développer des tests spécifiques et de les exécuter sur le composant BIT. Nous sommes en train de développer une extension de cette librairie visant à automatiser un certain nombre d'étapes du développement de l'utilisation de cette technologie. Enfin, afin d'assister au mieux le développeur d'applications basées composants, nous travaillons à intégrer ces deux axes de recherche dans un environnement cohérent et complet (méthode, outils, librairies, ...). Cet environnement traitera la composition logicielle des phases d'analyse et de conception jusqu'à la phase de test. Pour cela, en complément de nos travaux, nous suivons ceux menés par la communauté sur l'implémentation physique de l'assemblage de composants logiciels, en particulier dans les applications réparties. De plus, nous évaluerons la possibilité de générer automatiquement les tests permettant la validation de l'implémentation du composant par rapport aux spécifications définies à l'aide des modèles basés sur les types de relations Tout-Partie que nous aurons définis.

3. Bibliographie

- BELLOIR N., BRUEL J.M., « Intégration du test dans les composants logiciels », *Actes de la journée de travail du GDR I3*, Lyon, 2001.
- BELLOIR N., BRUEL J.M., BARBIER F., « Formalisation de la relation Tout-Partie : application à l'assemblage des composants logiciels », *Actes des journées composant : flexibilité du système au langage*, Besançon, 2001.
- HENDERSON-SELLERS B., BARBIER F., « Black and Whites Diamonds », *Proceedings of UML'99 conference*, LNCS 1723, Springer-Verlag, 1999.