

Résumé : Au cours de cette thèse, nous nous intéressons au test de logiciels orientés objet comme moyen principal pour la validation de composants. Cette étude se place dans le cadre d'une méthodologie pour la conception de composants logiciel fiables qui se base sur l'évaluation de la cohérence entre l'implantation, la spécification et les cas de test du composant. Les travaux présentés proposent des solutions pour le test fondées sur certaines spécificités de la programmation et de la conception orientée objet d'abord pour un composant puis pour un assemblage.

La qualité des cas de test étant le facteur principal pour la qualité du composant, la première contribution de cette thèse concerne la génération et l'optimisation automatique de cas de test. Nous étudions d'abord un algorithme génétique pour résoudre ce problème, puis nous proposons une adaptation originale de cet algorithme, plus efficace pour la génération de test et que nous appelons algorithme bactériologique.

Nous étudions ensuite l'impact de la conception par contrat sur la détection et la localisation des erreurs dans un assemblage de composants, à travers deux mesures : la robustesse et la diagnosabilité. Les résultats montrent que la simple introduction de contrats réduit rapidement l'effort de détection et de localisation des erreurs, et qu'au-delà d'une densité minimale, la qualité des contrats (leur capacité à détecter une erreur) est le facteur le plus important pour ces deux mesures.

L'utilisation de mécanismes tels que la délégation et le polymorphisme entraîne une forte répartition du contrôle à travers tout le système, et implique des interactions complexes entre les objets. La troisième contribution de cette thèse est l'étude de l'impact de ces structures de contrôle particulières pour la testabilité d'un assemblage. Nous définissons un critère de test et lui associons une mesure de testabilité, calculable à partir d'un diagramme de classes et qui permet d'évaluer l'effort de test dès la conception. Au cours de cette étude nous observons aussi l'impact de l'utilisation de design patterns sur la testabilité d'un système orienté objet. Nous montrons comment, en forçant une utilisation importante de l'héritage et de la délégation, ils suppriment certains problèmes de testabilité des programmes procéduraux mais introduisent d'autres difficultés qui peuvent être évaluées avec le critère de test proposé.