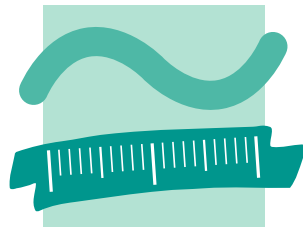


Exposé zur Masterarbeit

Variantenspezifische Abhängigkeitsregeln und Testfallgenerierung in TESTONA



**BEUTH HOCHSCHULE
FÜR TECHNIK
BERLIN**

University of Applied Sciences

Fachbereich VI - Technische Informatik - Embedded Systems



BERNER & MATTNER
AN ASSYSTEM COMPANY

Eingereicht von : Matthias Hansert
Matrikelnummer : s791744
Email-Adresse : matthansert@gmail.com

Inhaltsverzeichnis

Einführung	2
Motivation	2
Variantenspezifische Abhängigkeitsregeln	2
Workflow	2
Parametersetzung	3
Variantenspezifische Testfallgenerierung	3
Anhänge.....	5
Anhang 1	5
Anhang 2	5

Einführung

TESTONA¹ basiert auf der in der Praxis erfolgreich angewendeten Klassifikationsbaum-Methode². Es ist unabhängig von der Domäne und daher universell einsetzbar. Durch die Anwendung von TESTONA lassen sich Testspezifikationen schneller erstellen und leichter handhaben. Darüber hinaus profitieren Anwender von ständigen Erweiterungen und der Verbesserung bestehender Funktionen.

Motivation

Ziel dieser Masterarbeit ist die Verbesserung der Testfallgenerierung und der Testabdeckung bei mehreren Produktvarianten, die Ersetzung von Parameter, die Prozessoptimierung sowie die Handhabung für den Benutzer in der TESTONA-Umgebung. Jedes Produkt kann unterschiedliche Produktvarianten beinhalten und jede Variante besteht aus unterschiedlichen Komponenten mit unterschiedlichen Parametern. In Abhängigkeit von der ausgewählten Variante sollen bei der Testfallgenerierung die dazugehörigen Komponenten berücksichtigt werden und die erzeugten Testfälle dargestellt werden. Besonders zu beachten sind dabei die definierten Abhängigkeitsregeln sowie die darauf bezogene Testabdeckung.

Variantenspezifische Abhängigkeitsregeln

Abhängigkeitsregeln werden definiert um redundante Testfälle zu vermeiden, bzw. um Vorbedingungen für die Testfälle zu erstellen. Da Varianten verschiedene Bauelemente beinhalten, kann es dazu kommen, dass Bauelemente für Abhängigkeitsregeln nicht vorhanden sind. Dadurch könnte TESTONA bei der Testfallgenerierung die Testabdeckung verfälschen, indem die Gültigkeit eines Testfalles nicht garantiert werden kann. Um dieses Problem zu umgehen, muss bei der Erzeugung von Abhängigkeitsregeln auf mögliche Konflikte hingewiesen werden. Für den Lösungsansatz gibt es verschiedene Thesen die analysiert werden müssen, um eine optimale Prozessoptimierung zu erreichen.

Workflow

Um die Handhabung der Varianten bezogen auf die Testfälle und die Testgenerierung benutzerfreundlicher und effizienter zu gestalten, soll die Benutzung des Variantenmanagements durch einen Testingenieur untersucht werden. Resultierend aus den erworbenen Erkenntnissen wird das Lösungsdesign für eine Erweiterung des bestehenden Variantenmanagements in TESTONA konzipiert.

¹ Testdesign mit der Klassifikationsbaum-Methode

² Methode zur Ermittlung funktionaler Back-Box Tests

Parametersetzung

Einer der besonderen Eigenschaften von TESTONA ist die Kopplung mit Anforderungsspezifikationen die in IBM Rational DOORS³ definiert worden sind. Durch das DOORS Add-On MERAN können Anforderungen die in DOORS definiert sind, mit den zugehörigen Varianten verknüpft werden. Diese Varianten können in TESTONA eingebunden werden, durch eine erfolgreiche Anmeldung bei DOORS (über die TESTONA Oberfläche) und ein gezieltes Auswählen der gewünschten Varianten. Hierbei sollen die in den Anforderungen definierten Variablen (z.B. eine Geschwindigkeit oder Anzahl der Türen eines Autos) mit gespeichert werden. Im Klassifikationsbaum soll je nach ausgewählter Variante (z.B. der Name von Klassen) mit dem entsprechenden Wert ersetzt werden. Andere Lösungsmöglichkeiten werden noch untersucht.

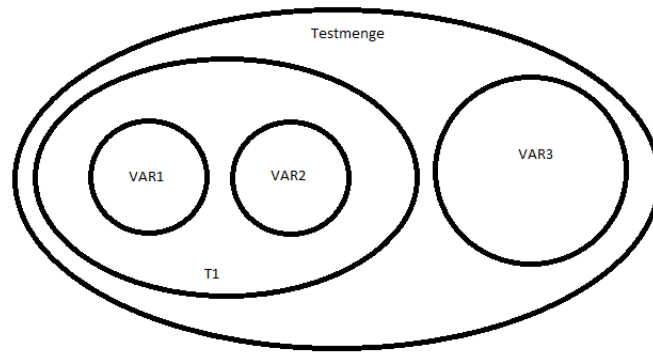
Variantenspezifische Testfallgenerierung

Der derzeitige Varianten-Management-Ansatz in TESTONA ist nicht in der Lage für die Testfallgenerierung zwischen verschiedene Varianten zu unterscheiden. Zwar werden durch die *Perspektive* „Variant Management“ verschiedene Varianten unterschieden, aber die Testfälle müssen manuell mit den jeweiligen Varianten verknüpft werden. Im Falle einer automatischen Testfallgenerierung werden auch ungültige Bauelemente betrachtet (siehe Anhang 1 und 2). Um dies zu vermeiden muss der Testingenieur einzelne Generierungsregeln anlegen. Dieser Vorgang soll automatisiert und von TESTONA übernommen werden.

Dabei gibt es verschiedene Betrachtungsweisen und mehrere Lösungswege. Entscheidend für die Lösung werden die erworbenen Kenntnisse über die Benutzung des Variantenmanagements durch einen Testingenieurs. Bei der Lösung ist zu beachten, dass eine komplette Testfallabdeckung garantiert werden muss.

Entspricht die „Testmenge“ aller Testfälle eines Produktes und „VAR1“, „VAR2“, „VAR3“ sind die Varianten dieses Produktes, so sind die Varianten und „T1“ Teilmengen von „Testmenge“. Dabei sind „VAR1“ und „VAR2“ echte Teilmengen von „T1“. Es muss berücksichtigt werden, dass die Darstellung nicht die Testgenerierung beeinflusst. Werden die Testfälle von „T1“, „VAR1“ und „VAR2“ generiert, so muss beachtet werden, dass für die Teilmengen „VAR1“ und „VAR2“ die Testfälle nicht neu generiert werden müssen, da diese schon in der Obermenge „T1“ beinhaltet sind. Betrachten wir die Darstellung der Varianten, so müssen unter „T1“ auch die Testfälle der Varianten „VAR1“ und „VAR2“ dargestellt werden, aber bei „VAR1“ und „VAR2“ nur die Schnittmenge von „T1“ mit der jeweilige Variante.

³ Dynamic Object Oriented Requirements System, Anforderungsmanagement Software



Durch diese Erweiterungen wird die Erstellung einer Testspezifikation weiter automatisiert. Dadurch lassen sich Fehler vermeiden und es kann Bearbeitungszeit eingespart werden.

Als Beispiel (siehe Anhänge) wird angenommen, eine Bilderkennungsfunktion besitze eine Variante „roter kreis“, in der rote Kreise erkannt werden sollen. Diese Variante enthält die Klassifikation „Farbe“ mit der Klasse „Rot“, die die Klassifikation „Tön“ $\in \{\text{Karmin, Rosa, Zinnober}\}$ umfasst. Die Variante enthält auch die Klassifikation „Form“, mit der Klasse „Kreis“ mit einer Klassifikation „Größe“ $\in \{\text{Klein, Mittel, Groß}\}$.

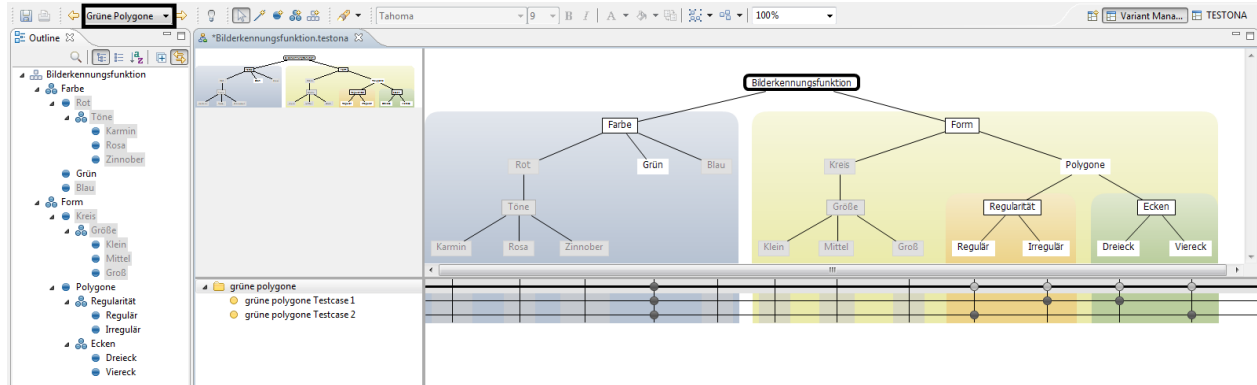
Die gleiche Bilderkennungsfunktion umschließt eine zweite Variante „grüne Polygone“ mit den Klassifikationen „Farbe“ und die Klasse „Grün“ und „Form“ und die Klasse „Polygone“. Die Klasse „Polygone“ beinhaltet die Klassifikationen „Regularität“ $\in \{\text{Regulär, Irregulär}\}$ und „Ecken“ $\in \{\text{Dreieck, Viereck}\}$.

Je nach ausgewählter Variante, sollten auch die aktuellen Parameter mit konkreten Werten aktualisiert werden. Bezogen auf das Beispiel soll die Variable „Größe“ der Variante „roter kreis“ der Durchmesser in cm darstellen (aus MERAN). Es wird davon ausgegangen, dass die Parameter in MERAN kompatibel zur Baumstruktur und Klassen/Klassifikationen sind.

Anhänge

Anhang 1

Korrekte Testfallgenerierung für die Variante „Grüne Polygone“



Anhang 2

Inkorrekte Testfallgenerierung für die Variante „Roter Kreis“. Bei der Testgruppe „Default Minimal Combination“ werden auf ungültige Elemente zugegriffen.

