Diplomarbeit
Thema : Reflexionsmethode für Produktlinien
Aufgabenstellung

Oscar Mourbare

11. Juni 2008

1 PL Abbildung im RFG

1.1 Base View

Für die interne Representation der Produktlinie, wird ein Base View erstellt. Dieses Base View wird alle Knoten und Kanten aller Varianten enthalten. Die Knoten werden eindeutige `Linkage.Name` haben, dafür wird der ursprungliche `Linkage.Name` umbenannt. Zum Umbenennen wird ein Präfix zu dem `Linkage.Name` hinzugefügt. Es wird auch für die Knoten ein neues Attribut `Variant.Name` hinzugefügt, das Information über die Knotenherkunft liefert. Ausserdem wird ein hierarchischer Knoten für jede Variante erstellt, der die Knoten und Kanten jeder Variante beinhaltet. Der hierarchische Knoten steht für weitere Verwendung zur Verfügung sowie für die Auswahl von Varianten, die man beobachten will.

1.2 View für Varianten

Für jede Variante wird ein View erstellt, das alle Knoten und Kanten der Variante enthält. Der Zweck dieses Views ist, die Quellcodeanalyse, die Architektursrekonstruktion und das Programmverstehen der entsprechenden Varianten zu ermöglichen. Sonst hättet man für jede einzelne Variante separat GraVis starten müssen. Man kann es direkt vom Produktlinie RFG ermöglichen.Um so weiter hin auf potentielle Fehler und Problemstellen der Variante hinzuweisen.

1.3 View Mapping

In Software Produktlinien ist die Wartung von Variabilitäten eine wichtige Aktivität. Das ist der Grund warum die Gemeinsamkeiten und Unterschiede zwischen den einzel-

nen Varianten beschrieben werden sollen. Es wird dafür ein View erstellt, das nur die Gemeinsamkeiten und Unterschiede zwischen Produkten beinhalten. Die Beziehungen zwischen den einzelnen Funktionen werden folgendersmaßen abgebildet

Identische Funktionen Identische Funktionen werden mit einer Kante des Typs `func_ident` zu repräsentanten einem Knoten `Ident_Function` verbunden.

Ähnliche Funktionen Ähnliche Funktionen werden mit einer Kante des Typs `func_sim` verbunden. Diese Kante wird zusätzliche Attribute haben, um die prozentuale Ähnlichkeit auszudrücken.

1.4 New View

Eine Software Produktlinie ist eine Menge von Varianten. Es besteht die Gefahr, dass man viel Varianten hat und will nur einige davon untersuchen. Deswegen wird eine nicht automatische Generierung eines Views möglich. Um so ein View zu erstellen, müssen die Varianten ausgewählt werden, von denen man die Informationen haben will. Die Informationen kann man sich auch aussuchen, beispielweise die identischen Funktionen der gewählten Varianten oder die ähnlichen Funktionen über 70 Prozent. Die View wird von diesen drei oben genannten Views generiert.

2 Ähnlichkeitsmaße

Die Software Produktlinie sind Charakterisiert mit ihren gemeinsamen Punkten aber auch mit ihren Unterschieden. Um die Gemeinsamkeiten und Unterschiede zwischen Software-Varianten herauszufinden, werden Ähnlichkeitsvektoren untersucht. Das ein einzelnes Maß vermutlich nicht genug ist wird eine Analyse aus den im Folgenden erwähnten Ähnlichkeitsmaßen zeigen.

2.1 Textbasiert

Hier handelt es sich um die Maße für die Unterschiede bzw. Ähnlichkeiten zwischen Zeichenketten. Diese Methoden zum finden von richtigen Werten sowie Levenshtein Distanz basierend auf der minimalen Anzahl an Operationen(Löschen, Einfügen, Ändern). Diese Methode ist notwendig zur Überführung von Zeichenkette (Funktion) in eine andere. Sie ist ein relativer Wert zwischen 0 und 1. Sie wird durch die Anzahl von Editierungsoperationen im schlechtesten Fall, d.h. zwei völlig unterschiedlichen Sequenzen geteilt. Eine andere Methode ist das Inference Basic, hierzu werden die Buchstaben einer Zeichenkette mit den Buchstaben einer andere Zeichenkette verglichen, in dem man immer in die gleiche Richtung geht. Es liefert die Anzahl von gefundenen Übereinstimmungen zwischen Quell- und Zielwort. Diese Methoden untersuchen, evaluieren, vergleichen ihrer Komplexität und zum Schluss entscheiden welche Methode richtig oder geeignet für die Produltlinie ist.

2.2 Ähnliche Verwendung/Verwendet

Ein anderes Ähnlichkeitmaß, das man berücksichtigen muss, ist wenn Funktionen, andere ähnliche Funktionen verwenden, oder wenn Funktionen von ähnlichen Funktionen verwendet werden handelt sich wahrscheinlich um Funktionen, die genauso ähnlich sind. Hier geht es auch darum festzustellen, wann zwei Funktionen ähnlich sind. Wenn mehrere Funktionen die gleiche Bibliotheken benutzen, können wir davon ausgehen, dass die Funktionen miteinander zu tun haben oder Ähnlich sind und diese möglicherweise mit einer Kante verbinden. Die Ähnlichkeit basiert dann auf der Abhängikeit von Komponenten oder Funktionen.

2.3 Schnittstellen-Ähnlichkeit

Die Schnittstellen stellen eine Art Vererbung dar, sie vereinbart gemeinsame Signaturen, welche in unterschiedlichen Komponenten implementiert werden. Somit können mehrere Komponenten, die gleiche Schnittstelle haben, und sogar wenn die Implementierung der Komponenten anders ist, kann eine gleiche Schnittstelle auf gleiche bzw. Ähnliche Funktionalität hinweisen. Wenn zwei Klassen/ Komponenten die gleiche Schnittstelle implementieren, dann haben vielleicht beide Funktionen in diesen Klassen, die gleiche Funktionalität.

3 Interaktive Abbildung

Die oben genannten Ähnlichkeitsmaße sind automatisierte Verfahren, und ersparen somit dem Software-Reengineer Handarbeit. Jedoch ist eine berechnete Ähnlichkeit nicht automatisch eine logische Korrespondenz zwischen verschiedenen Varianten. Das bedeutet dass die Ähnlichkeitsmaße nicht perfekt sind, die Ergebnisse enthalten Falsch-positive und Falsch-negative. Deswegen soll eine manuelle Alternative angeboten damit der Software-Reengineer entscheiden kann, welche Funktionen wirklich korrespondieren und so die Ergebnisse zu validieren. Dabei ist es auch zu beachten, dass wenn zwei Funktionen schon validiert wurden, will man es nicht nochmals validieren. Sollte man dann auch so gut wie möglicht sortieren: Beispiel in einem Fenster, die schon validierte Funktionen nach unten verschieben. Die noch nicht validierte Funktionen, die nicht sicher sind, nach Ähnlichkeitsprozent sortieren. Somit greift er auf die geschätze ähnlichen Funktionen zu um sie zu Validieren. Achten muss man auch auf Funktionen, die gar nichts miteinander zu tun haben aber von der Sicht des Reengineer ähnlich sind, er muss die Möglichkeit haben, neuen Beziehungen hinzuzufügen.