

# KABA – Klassen-Analyse mit Begriffsanalyse – Fallstudien (30 Min.)

Mirko Streckenbach  
Lehrstuhl Software-Systeme  
Universität Passau

Die Struktur einer Klassenhierarchie ist nicht immer optimal. Bei der Entwicklung einer Klassenbibliothek kann nicht vorhergesehen werden, wie diese später in Applikationen benutzt wird, zusätzlich kann sich ihre Struktur durch jahrelange Wartung und Erweiterungen von Programmen und Bibliothek noch verschlechtern. Typische Probleme sind:

- Bestimmte Members einer Klasse werden nur in einer bestimmten Unterklasse benutzt. Diese Members können in die Unterklasse verschoben werden.
- Bestimmte Members werden gar nicht mehr benutzt. Diese können entfernt werden.
- Verschiedene Instanzen einer Klasse benutzen verschiedene Members dieser Klasse. Die Klasse kann in verschiedene Klassen zerlegt werden.

Mit Hilfe von mathematischen Begriffsverbänden kann durch Analyse der Member-Zugriffe in objektorientierten Programmen eine Klassen-Hierarchie in einer neue transformiert werden. Diese neue Hierarchie ist verhaltensäquivalent, aber minimal, d.h. jede Klasse enthält nur die minimal notwendige Anzahl von Members.

Durch diese neue Hierarchie erhält man exakte Informationen, wie eine Klassenbibliothek von einem Programm benutzt wird. Die oben genannten Probleme werden automatisch beseitigt.

KABA ist eine Prototyp-Implementierung dieses Verfahrens. KABA analysiert Java Classfiles, ermittelt alle notwendigen Informationen zur Bestimmung des Begriffsverbandes und berechnet diesen. Der Begriffsverband wird graphisch dargestellt und kann bearbeitet werden.

Mit Hilfe von KABA war es möglich, eine Reihe von Fallstudien mit kleinen bis mittleren Programmen durchzuführen. Der Vortrag soll einige dieser Beispiele vorstellen und die Erkenntnisse, die daraus gewonnen werden können näher beleuchten.

Die erste Abbildung zeigt die ursprüngliche Klassenstruktur der benutzerdefinierten Klassen eines “Türme von Hanoi” Programms mit GUI Bibliothek (45 Klassen, 3000 Zeilen). Die zweite Abbildung zeigt die mit KABA gewonnene, neue Klassenstruktur. Die starke Verästelung zeigt wie hochgradig detailliert die Analyse ist, für nahezu jede Variable wird ein individueller Typ berechnet. Abbildung 3 zeigt die neue Struktur ohne explizite Typen für Pointer, was den Umfang der Hierarchie erheblich reduziert und es somit einfacher macht, Informationen daraus zu gewinnen.

Was für Erkenntnisse aus der neuen Hierarchie gewonnen werden können, soll an einem Beispiel aufgezeigt werden. Abbildung 4 zeigt eine Vergrößerung des Zweiges rechts außen von Abbildung 2. Variablen sind unterhalb und Members, die im neuen Typ einer Variable enthalten sind, überhalb eines Knotens

dargestellt. Die entsprechenden Original-Klassen sind die Vierer-Gruppe in Abbildung 1 links oben. Eine nähere Betrachtung zeigt, daß alle Typen oberhalb des "Mittelknotens" in Abbildung 4 keine Implementationen beinhalten und somit abstrakte Klassen sind. Die Knoten weiter unten beinhalten alle Implementationen. Sieht man von den Pointern ab, hat sich die Original-Struktur reproduziert. Dies weist auf einen gelungenen Entwurf bei diesen Klassen hin.

Die Struktur im mittleren Teil von Abbildung 2 ist wesentlich komplexer. Trotzdem lassen sich auch hier Erkenntnisse über die Benutzung der Original-Klassen gewinnen.

Die bereits analysierten Java Beispiele waren im Allgemeinen alle gut strukturiert. Vorgesehen ist, dieses Verfahren auch zur Analyse alter C++ Programme einzusetzen; hier sich sicherlich andere Ergebnisse zu erwarten und größeres Potential für Reengineering zu finden.

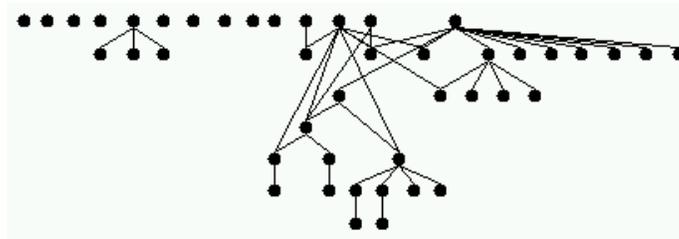


Abbildung 1: Original-Klassen-Hierarchie

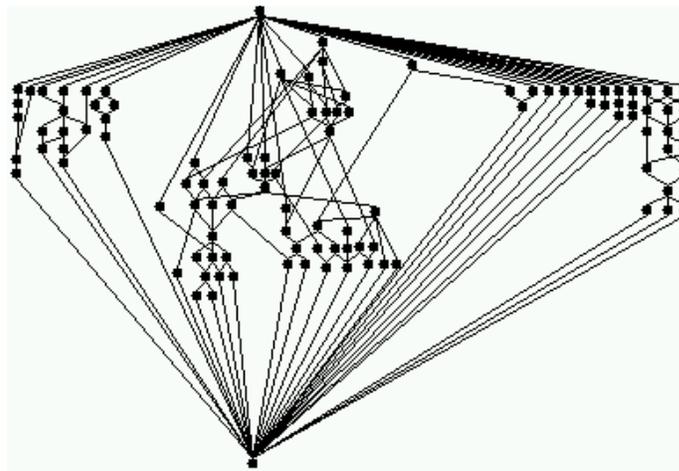


Abbildung 2: Neue Klassen-Hierarchie

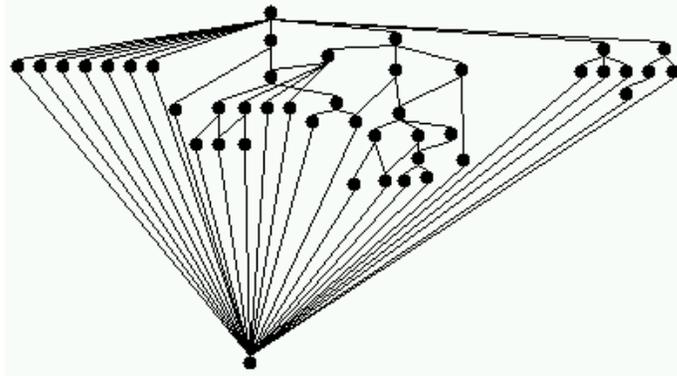


Abbildung 3: Neue Klassen-Hierarchie ohne Pointer

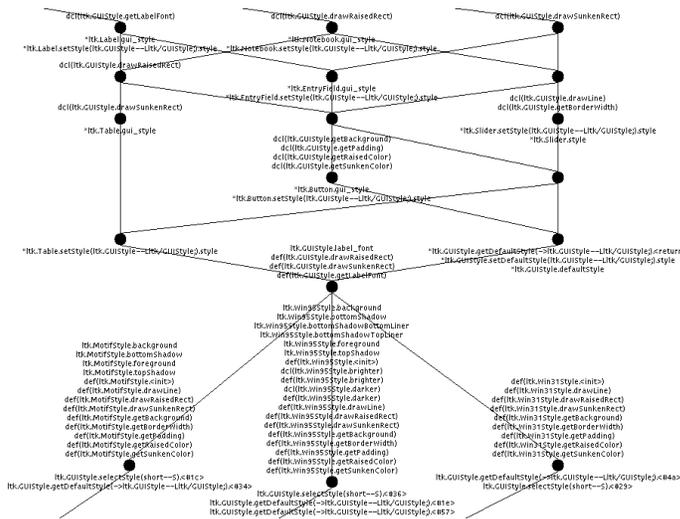


Abbildung 4: Detail Vergrößerung

### Literatur

- Bögemann, A., Streckenbach M.: Reengineering Class Hierarchies using Concept Analysis. Diplomarbeit, TU Braunschweig, März 1999.
- Snelting, G., Tip, F.: Reengineering Class Hierarchies Using Concept Analysis. Proc. ACM SIGSOFT Symposium on the Foundations of Software Engineering, November 1998, 99-110.
- Tip, F., Sweeney, P.: Class Hierarchy Specialization. Proceedings of the Twelfth Annual Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA'97). 1997.