



**Fachbereich 3  
Mathematik und Informatik**

**Exposee**

**Erstellen eines Eclipse-Plugins zur  
Visualisierung von  
Quelltextduplikaten**

Zur Diplomarbeit von  
Oliver Herdin

# 1. Motivation

**Gravis** enthält zwar einfache Möglichkeiten zur Visualisierung von Quelltextduplikaten (innerhalb eines Softwareprojektes kopierter Quellcode, kurz Klon genannt), aber keine komplexeren Programmierhilfen wie **Syntax-Highlighting** oder **Auto-Vervollständigung**, wie sie heutzutage von einer zeitgemäßen **IDE** erwartet werden. Dadurch ist sie als Entwicklungsumgebung für die eigentliche Programmierarbeit ungeeignet.

Dem Programmierer bleibt somit nur die Möglichkeit, während seiner Arbeit immer wieder aus seiner **IDE** nach **Gravis** zu wechseln, um sich aus der Klonerkennung gewonnene Informationen an zu schauen, oder er verzichtet während seiner normalen Programmier-tätigkeiten ganz darauf. In beiden Fällen wäre eine Representation von erkannten Quelltextduplikaten direkt in seiner **IDE** jedoch die bessere Lösung.

## 2. Definitionen

Ein Klonpaar besteht aus zwei identischen Quelltextabschnitten. Der Einfachheit halber gehen wir hier erst einmal von kompletter Gleichheit aus. Die duplizierten Quelltextabschnitte werden dabei als Codefragment bezeichnet, und können überall im Quelltext eines Projektes auftauchen, sowohl in der selben Datei als auch außerhalb. Innerhalb dieses Dokumentes wird die Bezeichnung Klon als Synonym für ein dupliziertes Codefragment genutzt.

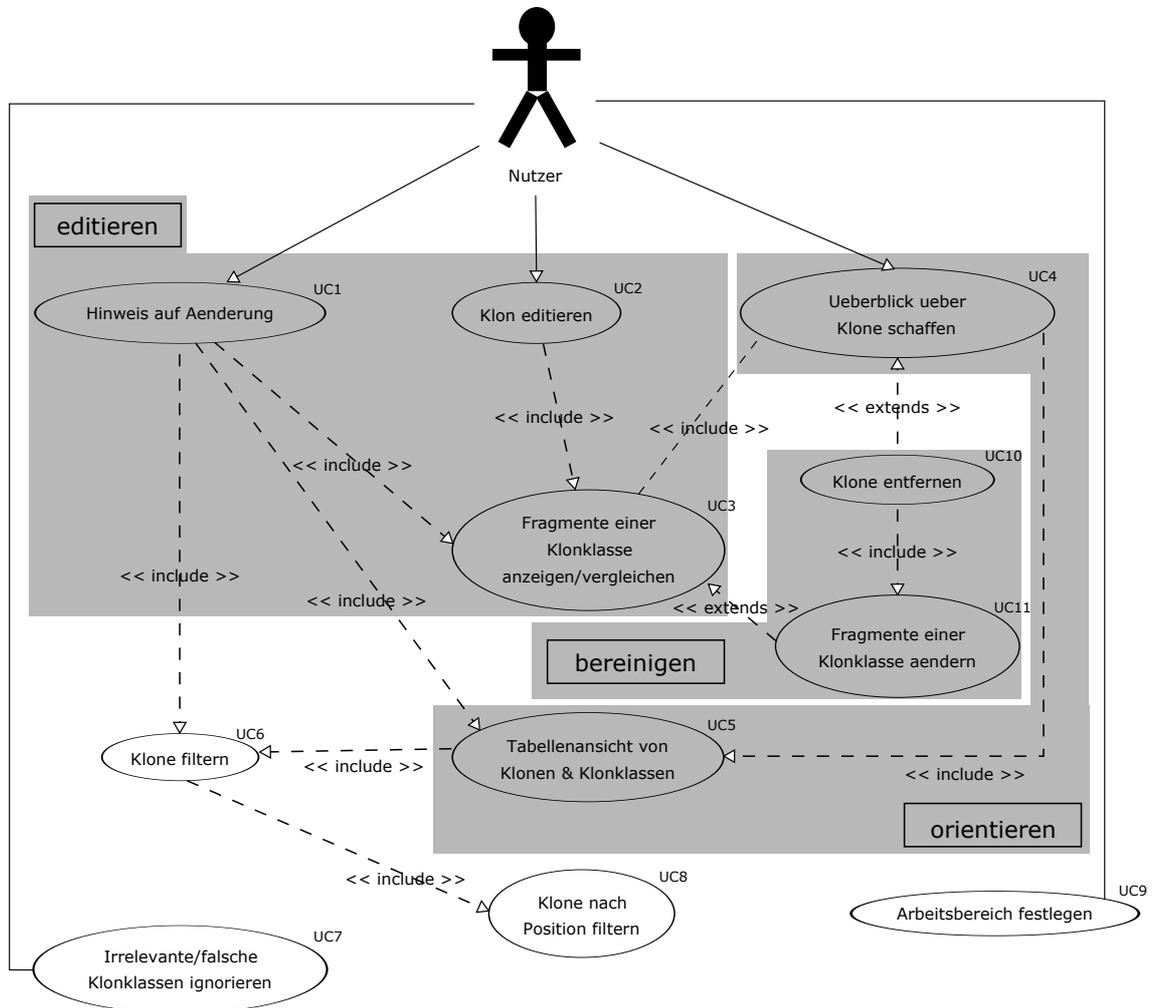
Eine Klonklasse besteht im Gegensatz zu einem Klonpaar aus einer beliebigen Zahl von identischen Codefragmenten, aber logischerweise aus mindestens zwei.

## 3. Anwendungsfälle

Der folgende Abschnitt befasst sich mit den Anwendungsfälle, an denen die durch das Plugin zu leistenden Funktionen festgelegt werden. Als erstes wird ein kurzer Überblick über die Anwendungsfälle in schriftlicher und folgend in grafischer Form als UML Diagramm geliefert. Danach werden die Anwendungsfälle ausführlicher erläutert.

Die Anwendungsfälle für den Programmierer, den Nutzer des Plugins, teilen sich in drei Gruppen auf: *editieren*, *orientieren* und *bereinigen*. In den Bereich *editieren* fallen Aktionen, die vom Programmierer während der täglichen Arbeiten am Projektquelltext ausgeführt werden, wie das einfache Editieren von Quelltext, oder das Nachbearbeiten von normalen Änderungen durch dritte Programmierer, die im selben Projekt tätig sind, und auf den eigenen Quellcode Auswirkungen haben. Der zweite Tätigkeitsbereich *orientieren*

befasst sich mit Anwendungsfällen, die dem Verständnis des Programmierers über das Projekt behilflich sind. In der letzten Gruppe der Tätigkeiten, *bereinigen*, befasst sich der das System nutzende Programmierer mit dem Entfernen von Klonen und Klonklassen, mit dem Ziel, die Qualität des Softwareprojektes zu steigern.



In jedem der folgenden Anwendungsfälle ist der Akteur jeweils nur der das System, in diesem Fall das Plugin, nutzende Programmierer.

### 3.1. Editieren

#### *UC1 Hinweis auf Änderung*

- Ziele:
  - Inkonsistente Änderungen an Klonen einer Klonklasse verhindern.
  - Durch neue Klone und Klonklassen erzeugte Abhängigkeiten frühzeitig erkennen.
- Zusammenfassung:

Wann immer sich ausserhalb der aktuellen Arbeitssitzung des Programmierers der Quelltext eines Klones ändert, wird der Programmierer auf die den Klon beinhaltende Klonklasse aufmerksam gemacht.
- Auslöser:

Eclipse aktualisiert die im aktuellen Projekt geöffneten Dateien, entweder weil Eclipse gerade gestartet wurde, weil das aktuelle Projekt gerade geöffnet wurde, oder weil der Nutzer die Aktualisierung von Hand ausgelöst hat.
- Ablauf:
  1. Alle Dateien, die sich außerhalb der aktuellen Arbeitssitzung geändert haben, werden auf nicht durch Anwendungsfall UC6 ausgefilterte Klone überprüft.
  2. Alle gefundenen Klone, in denen Änderungen vorgenommen wurden, werden in der durch Anwendungsfall UC5 spezifizierten Übersicht über Klone und Klonklassen hervorgehoben, so die zugehörige Ansicht geöffnet ist.
  3. Auf Wunsch kann zu Anwendungsfall UC3 übergegangen werden, um eine der Klonklassen genauer zu inspizieren und gegebenenfalls zu editieren. Dieser Schritt kann beliebig oft mit beliebigen Klonklassen aus der gefundenen Menge wiederholt werden.
- Eingefügte Anwendungsfälle:

UC6 Klone filtern, UC5 Listenansicht von Klonen & Klonklassen, UC3 Fragmente einer Klonklasse anzeigen/vergleichen

*UC2 Klon editieren*

- Ziele:
  - Schon vor der Änderung des eigentlich Quelltextes wissen, was einen an Mehraufwand entstehen kann, sollte der Quelltext ein Klon sein.
  - Die Duplikate eines Klones einfach inspizieren/editieren um bei Bedarf Inkonsistenzen zu beheben, nachdem Änderungen daran vorgenommen wurden.
- Zusammenfassung:

Die Position von Quelltextduplikaten wird innerhalb von offenen Editorfenstern markiert, und bei Bedarf kann die gesamte Klonklasse zu dem markierten Klon inspiziert und bearbeitet werden.
- Auslöser:

Öffnen einer Datei, in der Klone vorhanden sind.
- Ablauf:
  1. Sobald in einem offenen Editorfenster Quelltext, der zu einem Klon gehört, sichtbar wird, wird dieser markiert.
  2. Der Klon kann normal im Editor bearbeitet werden, dieser Schritt kann übersprungen werden.
  3. Mittels Anwendungsfall UC3 kann nun die Klonklasse dieses Klons inspiziert und/oder editiert werden. Nach belieben kann Punkt 2. und 3. wiederholt werden.
- Eingefügte Anwendungsfälle:

UC3 Fragmente einer Klonklasse anzeigen/vergleichen

### 3. Anwendungsfälle

#### *UC3 Fragmente einer Klonklasse anzeigen/vergleichen*

- Ziele:
  - Alle Fragmente einer Klonklasse inspizieren.
  - Ein Klon der Klonklasse allen anderen Fragmenten gegenüberstellen, zum Beispiel wenn sich ein Klon der Klasse verändert hat.
- Zusammenfassung:

Der Anwendungsfall beschreibt die Nutzung eines Editors, der selbst zwei Quelltexteditoren beinhaltet, um die Inspektion einer ganzen Klonklasse zu ermöglichen. Die beiden Editoren können dabei mit den Dateien, die die einzelnen Klone der Klonklasse enthalten, belegt werden, um einen Vergleich zwischen zwei Quelltextduplikaten und ihrer Umgebung zu ermöglichen.
- Auslöser:

Ein Klick auf den zugehörigen Button öffnet den Vergleichseditor für die zuvor in Anwendungsfall UC1 oder UC2 ausgewählte Klonklasse.
- Ablauf:
  1. Nachdem der Vergleichseditor geöffnet wurde befinden sich in den darin enthaltenen Editoren jeweils eine Datei, in der zumindest ein Klon der ausgewählten Klonklasse ist. Sollte die Klonklasse einen durch die letzten Änderungen veränderten Klon enthalten, oder kurz zuvor ein Fragment aus der Kloneklasse entfernt wurden sein, so wird dieses Fragment in einem der Editoren dargestellt, um den Grund für die Anpassung der anderen Quelltextduplikate gleich sichtbar vor sich zu haben.
  2. Mittels Vor- und Zurück-Buttons für jedes der beiden Fenster kann in dem jeweiligen Editor durch die verbleibenden Klone der Klonklasse navigiert werden. Dabei ersetzt die/der neue Datei/Klon immer die vorherigen Anzeige.
  3. Anwendungsfall-Erweiterung: UC11 Fragmente einer Klonklasse ändern.
- Eingefügt in Anwendungsfällen:

UC1 Hinweis auf Änderung, UC2 Klon editieren
- Erweitert durch Anwendungsfall:

UC11 Fragment einer Klonklasse ändern

## 3.2. Orientieren

### *UC4 Überblick über Klone schaffen*

- Ziele:
  - Verständniss für den Grad der Kopplung innerhalb des Projektes entwickeln.
  - Grund für Klone durch Inspektion ermitteln.
- Zusammenfassung:

In einer Tabellendarstellung aller Klone und Klonklassen, die für den Nutzer gerade von Interesse sind, können die Klonklassen und Klone nach beliebigen Kriterien sortiert werden. Bei Bedarf kann ein Klon oder eine Klonklasse auch direkt ausgewählt und inspiziert werden.
- Auslöser:

Durch Interaktion mit der in Anwendungsfall UC5 beschriebenen Ansicht.
- Ablauf:
  1. Nach dem Öffnen der in Anwendungsfall UC5 beschriebenen Tabellenansicht aller Klone und Klonklassen können die von diesen angezeigten Eigenschaften nach ihrer Wichtigkeit sortiert werden.
  2. Für jede Eigenschaft von Interesse wird die Sortierreihenfolge festgelegt, aufsteigend oder absteigend.
  3. Die Einträge der Anzeige werden entsprechend den Eingaben von 1. und 2. sortiert.
  4. a) Durch Doppelklick auf einen Klon wird die den Klon enthaltende Datei im normalen Editor geöffnet.  
b) Durch Doppelklick auf eine Klonklasse wird für diese Klonklasse der Anwendungsfall UC3 ausgeführt.
  5. Anwendungsfall-Erweiterung: UC10 Klone entfernen
- Eingefügte Anwendungsfälle:

UC3 Fragmente einer Klonklasse anzeigen/vergleichen, UC5 Listenansicht von Klonen & Klonklassen
- Erweitert durch Anwendungsfall: UC10 Klone entfernen

### 3. Anwendungsfälle

#### *UC5 Tabellenansicht von Klonen & Klonklassen*

- Ziele:
  - Eine Representation von Klonen und Klonklassen außerhalb des Quelltextes darstellen.
- Zusammenfassung:

Eine Ansicht, in der die im Projekt vorhandenen Klone und Klonklassen tabellarisch aufgeführt werden. Die Tabellen enthalten ausser die Position und Bezeichner auch noch andere Metriken.
- Auslöser:

Die Ansicht, die die Tabellen enthalet wird über den Eclipse eigenen Dialog zum öffnen von neuen Fenster geöffnet.
- Ablauf:
  1. Sobald geöffnet zeigt die Ansicht zwei Tabellen an, eine für Klonklassen und eine zweite für Klone.
  2. Die anzuzeigenden Klone und Klonklassen werden durch den Anwendungsfall UC6 gefiltert.
- Eingefügt in Anwendungsfällen:

UC1 Hinweis auf Änderungen, UC4 Überblick über Klone schaffen
- Eingefügte Anwendungsfälle:

UC6 Klone filtern

*UC6 Klone filtern*

- Ziele:
  - Für die aktuelle Arbeit unnötige Klone und Klonklassen ausfiltern
- Zusammenfassung:

Durch den Filter werden nur Klone und Klonklassen gelassen, die für die aktuelle Arbeit interessant sind.
- Auslöser:

UC1 Hinweis auf Änderung oder UC6 Tabellenansicht von Klonen & Klonklassen.
- Ablauf:
  1. Wenn andere Anwendungsfälle das filtern von Klonen und Klonklassen erfordern, werden alle Klonklassen des Projektes gegen die in Anwendungsfall UC7 ignorierten Klonklassen verglichen.
  2. Alle übrig gebliebenen Klone und Klonklassen werden nach Anwendungsfall UC8 auf ihre Position im Projekt überprüft und gegebenenfalls aussortiert.
- Eingefügt in Anwendungsfällen:

UC1 Hinweis auf Änderungen, UC5 Tabellenansicht von Klonen & Klonklassen
- Eingefügte Anwendungsfälle:

UC8 Klone nach Position filtern

### 3. Anwendungsfälle

#### *UC7 Irrelevante/falsche Klonklassen ignorieren*

- Ziel:
  - Klonklassen, die falsch erkannt wurden ignorieren
  - Klonklassen, die irrelevant sind ignorieren
- Zusammenfassung:

Der Nutzer selektiert die für ihn irrelevanten Klonklassen, so dass sie vom Plugin nicht mehr weiter beachtet werden.
- Auslöser:

Eine Klonklasse wurde gefunden, die ignoriert werden soll
- Ablauf:
  1. Der Nutzer wählt die zu ignorierende Klonklasse aus.
  2. Mittels eines Buttons oder Menüpunktes wird die Klonklasse zur Liste der zu ignorierenden Klonklassen hinzugefügt.

*UC8 Klone nach Position filtern*

- Ziel:
  - Klone und Klonklassen, die nicht in einem bestimmten Bereich im Dateisystem des Projektes liegen, ausfiltern.
- Zusammenfassung:  
Klone und Klonklassen werden auf ihre Position im Projekt hin überprüft, und gegebenenfalls ausgefiltert
- Auslöser:  
UC6 Klone filtern
- Ablauf:
  1. Alle zu filternden Klone werden daraufhin überprüft, ob sie in dem in UC9 festgelegte Arbeitsbereich im Projekt liegen.
  2. Wenn nicht, wird der Klon ausgefiltert.
  3. Sollten alle Klone einer Klonklasse ausgefiltert wurden sein, so wird auch die Klonklasse ausgefiltert.
- Eingefügt in Anwendungsfällen:  
UC6 Klone filtern

### 3. Anwendungsfälle

#### *UC9 Arbeitsbereich festlegen*

- Ziel:
  - Die Ordner und Dateien im Projekt fest legen, in denen sich der Nutzer während seiner aktuellen Arbeit bewegt, seinen Arbeitsbereich.
- Zusammenfassung:

Durch Selektion von Dateien und Ordnern wird der aktuelle Arbeitsbereich des Entwicklers festgelegt
- Auslöser:

Den Nutzer interessiert aktuell nur ein Teil des gesamten Projektes
- Ablauf:
  1. Der Nutzer markiert den für ihn interessanten Bereich innerhalb des normalen Eclipse Dateinavigators.
  2. Die markierten Dateien und Ordner werden über ein Kontextmenüeintrag dem aktuellen Arbeitsbereich hinzugefügt.
  3. Der aktuelle Arbeitsbereich kann unter einem beliebigen Bezeichner gespeichert werden.
- Alternativer Ablauf:
  1. Aus der Liste der gespeicherten Arbeitsbereiche wird einer ausgewählt und aktiviert als momentaner Arbeitsbereich.

### 3.3. Bereinigen

#### *UC10 Klone entfernen*

- Ziel:
  - Die Qualität des Quelltextes erhöhen, durch die Reduktion von Klonen.
- Zusammenfassung:

Den Basis-Anwendungsfall UC4 Überblick über Klone schaffen erweitert dieser Anwendungsfall, indem er die Klone nicht nur inspiziert, sondern durch bearbeiten entfernt.
- Auslöser:

In UC4 wird ein Klon oder eine Klonklasse gesichtet, die entfernt werden soll.
- Ablauf:
  1. bis 3. und 4b. von UC4
  2. Die im Vergleichseditor geöffnete Klonklasse wird dem Anwendungsfall UC11 nach durch editieren der Codefragmente der Klonklasse entfernt.
- Erweiterte Anwendungsfälle:

UC4 Überblick über Klone schaffen
- Eingefügte Anwendungsfälle:

UC11 Fragmente einer Klonklasse ändern

### 3. Anwendungsfälle

#### *UC11 Fragmente einer Klonklasse ändern*

- Ziel:
  - Die Klone der im Vergleichseditor geöffneten Klonklasse zu bearbeiten
- Zusammenfassung:

Den Basis-Anwendungsfall UC3 Fragmente einer Klonklasse anzeigen/vergleichen erweitert dieser Anwendungsfall, indem die geöffneten Dateien nicht nur inspiziert sondern auch editiert werden.
- Auslöser:

Einer der Anwendungsfälle, die den Basis-Anwendungsfall UC3 einfügt macht das editieren der geöffneten Klonklasse nötig, oder UC10 Klone entfernen.
- Ablauf:
  1. und 2. von UC3
  2. In den beiden Editoren des Vergleichseditor kann ganz normal editiert werden, wie in jedem anderen Editor in Eclipse auch. Die, durch die zu diesem Anwendungsfall führenden Fällen, nötig gewordenen Änderungen können vom Nutzer normal durchgeführt werden.
- Erweiterte Anwendungsfälle:

UC3 Fragmente einer Klonklasse anzeigen/vergleichen
- Eingefügt in Anwendungsfall:

UC10 Klone entfernen