

# Konzeption und Entwicklung eines Auswertungstools für die Logdateien des JRockit Garbage Collectors

Raffael Schmid

Hochschule für Technik Zürich, Studiengang Informatik (1. Dezember 2011)

Garbage Collection nennt man eine Methode zur automatischen Speicherbereinigung von nicht mehr verwendeten Objekten. Sie findet Anwendung in allen modernen Programmiersprachen, ist in der Regel sehr Ressourcenintensiv und hat Einfluss auf das Laufzeitverhalten der Applikation. Ob Performanceprobleme im Zusammenhang mit der Garbage Collection stehen, kann beispielsweise mit der Suche nach dem Dominating Consumer (dominierender Ressourcenverbraucher) [1] eruiert werden. Die Detailanalyse wird anschliessend auf der Basis von Garbage Collection Logdateien oder mit Werkzeugen zur Analyse am laufenden System gemacht. Log-Analysewerkzeuge sind bereits für einige virtuelle Maschinen (Laufzeitumgebung) verfügbar. Diese Arbeit konzipiert und implementiert eine Software zur Automatisierung der Analyse der Garbage Collection Logdateien der JRockit Virtual Machine (Release R28).

Die Evaluation von Rich Client Framework (Bibliothek zur Implementation von Desktop-Anwendungen) und Charting-Bibliothek (zur Anzeige von Diagrammen) sowie die Ermittlung der funktionalen Anforderungen ist getrieben durch eine methodenbasierte Anforderungsanalyse. Es standen drei Methoden zur Auswahl: Use Cases, Requirements Engineering nach IEEE 830 und User Stories. Auf Ebene der Customer Requirements wird die Methode der Use Cases eingesetzt. Sie zeichnet sich dadurch aus, dass man die Anforderungen nicht nur textuell sondern auch modellbasiert mit UML (Modellierungssprache) definieren kann. Für die Dokumentation der Development-Requirements, beinhaltet auch die Qualitätsanforderungen, wird der Standard IEEE 830 verwendet. Die damit definierten Anforderungen lassen sich gut in einzelne Entwicklungspakete (Features) verpacken. Auch die Struktur der Anforderungsanalyse wurde anhand dieser Definition aufgebaut.

Die Analysesoftware basiert auf dem Rich Client Framework Eclipse 3.x. Für die Anzeige der Diagramme wird JFreeChart verwendet. Die Software wird als Plugin für die Entwicklungsumgebung bereitgestellt. Die Installation kann über das Netzwerk durch Angabe einer Software-Seite gemacht werden, auch Updates können im laufenden Betrieb durchgeführt werden. Der Anwender startet eine Analyse, indem er eine Logdatei aus dem Dateisystem importiert. Er ist nun in der Lage, das Analysefenster für die importierte Logdatei zu öffnen. Durch die Applikation wird der Inhalt der Datei geladen, mittels Regulären Ausdrücken in Tokens umgewandelt (Lexer, lexikalischer Scanner), semantisch analysiert und in eine strukturierte Form (Objektstruktur) umgewandelt. Die Verarbeitung der Daten ist in eine separate Erweiterung für die Analysesoftware ausgelagert. Somit können zu einem späteren Zeitpunkt auch andere Erweiterungen entwickelt werden. Das geöffnete Analysefenster zeigt die Daten in unterschiedlichen Ansichten. Der Benutzer kann in der Standardanalyse die statistische Übersicht, der genutzte Speicher über die Zeit und die Dauer jeder einzelnen Garbage Collection anschauen. Es besteht aber auch die Möglichkeit, aus den verfügbaren Daten eigene Diagramme zu erstellen. Verschiedene benutzerdefinierte Diagramme können zusammen als ein Benutzerprofil gespeichert werden. Es besteht auch die Möglichkeit, diese Profile zu exportieren, um anderen Benutzern zugänglich zu machen. Zur Benutzerführung stehen ein Dashboard (Übersicht aller verfügbaren Aktionen und Hilfethemen) sowie verschiedene Hilfeseiten zur Verfügung. Die Texte (Labels) und Hilfeseiten sind grösstenteils in Deutsch und Englisch verfügbar. Sie werden je nach Eclipse-Spracheinstellung geladen.

Mit dem Prototyp der Analysesoftware können die Garbage Collection Logs der JRockit Virtual Machine (Release 28) hinsichtlich verschiedener Kenndaten analysiert werden. Der Anteil an manueller Arbeit kann somit verringert werden. Aktuell können für Release 27, der noch vielerorts im Einsatz ist, keine Auswertung gemacht werden. Dies wäre für den realen Einsatz wünschenswert. Zudem kann der Logger der Virtual Machine noch feiner eingestellt werden, so dass er weitere Informationen liefert. Diese wären aber in einer Analyse mit dieser Software noch nicht ersichtlich.

[1] Kirk Pepperdine. Concurrent and performance reloaded.

<http://www.jfokus.se/jfokus/page.jsp?id=recordings#page=page-1>, 2011.