

Generierte Anwendung modernisieren – Reengineering für Fortgeschrittene?

Daniela Schilling
dschilling@delta-software.com
Delta Software Technology GmbH

Abstract

Generatoren haben in der Software-Entwicklung eine lange Historie. Bis heute werden Generatoren aus den 1970er und 80er Jahren eingesetzt, um große und komplexe Anwendungen zu erzeugen. Durch ihre lange Lebensdauer haben auch generierte Anwendung Modernisierungsbedarf. Allerdings ist es hierfür notwendig, Reengineering-Methoden auf den Generator anzuwenden, statt auf die generierte Anwendung, wodurch neue Herausforderungen an den Ansatz entstehen.

1 Bekanntes Problem, andere Ebene

In den 1970er und 80er Jahren kamen einige Generatorsysteme für COBOL auf den Markt. Mit diesen Systemen haben vor allem Banken und Versicherungen ihre großen und komplexen Kernanwendungen entwickelt und pflegen diese Anwendung unter Nutzung der Generatorsysteme bis heute.

Die Anwendungen verrichten immer noch zuverlässig ihren Dienst. Allerdings wurden sie im Laufe der Zeit von Entwickler zu Entwickler weitergeben, an neue Anforderungen angepasst und für neue Aufgaben erweitert. Und so gilt für diese Anwendungen das Gleiche, was auch für native Anwendungen nach vielen Jahrzehnten im Einsatz gilt:

- über die Zeit haben sich technische Schulden angesammelt, die die Wartung des Generators erschweren,
- Laufzeit und Performance entsprechen nicht mehr den Anforderungen
- Teile der Anwendung sollen neugeschrieben oder in Standardprodukte ausgelagert werden, was jedoch die aktuelle Architektur der Anwendung nicht zulässt.

Kurz gesagt ein Reengineering ist erforderlich. Damit der Generator auch weiterhin eingesetzt werden kann, reicht es jedoch nicht aus das Reengineering auf der generierten Anwendung auszuführen,

stattdessen muss der Input für den Generator modernisiert werden.

2 Beispiel Delta ADS

Seit Mitte der 1970er Jahre bietet Delta das Generatorsystem ADS für die Generierung von COBOL- und PL/I-Anwendungen. Das System setzt sich zusammen aus Prozessoren und Standard-Macros. ADS bietet eine DSL zur Spezifikation der Kundenanwendungen. Programme spezifizieren die Abläufe und die Anwendungslogik, mittels Macros können wiederkehrende Anteile spezifiziert werden.

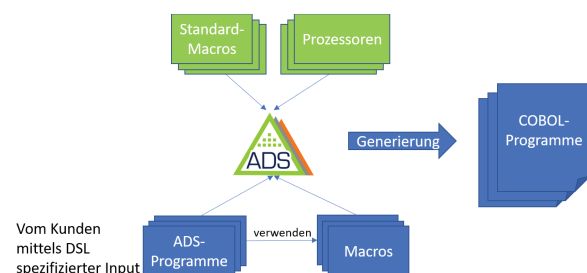


Figure 1: Delta ADS

Bis heute werden mit ADS mehr als 100 Mio Lines of Code generiert.

3 Herausforderung Reengineering von Generatoren

Beim Einsatz von Generatoren gilt häufig, dass der Input für den Generator eine möglichst hohe Les- und Wartbarkeit aufweisen soll, während der generierte Code eher in Bezug auf Performance etc. optimiert wurde. Dennoch kann es bei einem Reengineering notwendig werden, auch den generierten Code zu betrachten. Soll der Generator aber weiterhin eingesetzt werden, reicht es nicht aus, nur den generierten Code zu überarbeiten. Denn, bei einer erneuten Generierung würden alle Änderungen verloren gehen.

Welche Anteile betrachtet werden müssen, hängt ganz entschieden von der Aufgabe ab. Hat die Effizienz bei der Wartung nachgelassen, so muss der Generator-Input überarbeitet werden. Besonders spannend ist jedoch der Fall, dass die generierten Programme nicht mehr den Anforderungen entsprechend und einem Redesign unterzogen werden sollen. In diesem Fall gilt es, die Programme zu analysieren und die Ergebnisse auf den Generator-Input zurück zu führen, um dort die notwendigen Änderungen vorzunehmen.

4 Gemeinsam zum Ziel

Um eine generierte Anwendung modernisieren zu können, ist unterschiedliches Wissen erforderlich. Zum einen muss bekannt sein, welche Modernisierungen vorgenommen werden sollen und an welchen Stellen der generierten Anwendung diese beispielhaft anzuwenden sind. Zum anderen wird Wissen über den Generator benötigt, insbesondere darüber welche Artefakte für die Generierung von bestimmtem Code verantwortlich ist und wie ggf. für die Generierung notwendige Parameter belegt waren.

4.1 Modernisierung einer generierten Anwendung

Eine Kunde möchte seine mit Delta ADS generierte COBOL-Anwendung modernisieren. Dazu haben der Kunde und Delta ein gemeinsames Vorgehen entwickelt.

Der Kunde hat das notwendige Fachwissen und legt fest, was modernisiert werden soll. Beispiel: Im generierte Programm P1 sollen die Zeilen 27-40 überarbeitet werden. Im ersten Schritt ist zu prüfen, aus welchen Artefakten die Zeilen generiert wurden. ADS erzeugt dafür bei der Generierung Back-Referenzen, d.h. Informationen im generierten Sourcecode, aus welchem Artefakt und welcher Zeile die jeweilige Zeile generiert wurde. Damit kann die relevante Stelle gefunden werden. Delta hat, als Hersteller von ADS, zusätzlich die Möglichkeit auch die Belegung der Generierungsparameter auszugeben. Im Beispiel stammen die Zeilen aus dem Macro M1.

Aber Achtung: Macros dienen der Wiederverwendung, d.h. sie können beliebig oft zur Generierung angezogen werden, sowohl innerhalb des Programms P1 als auch in weiteren Programmen. Bevor also das Macro modifiziert werden kann, muss festgestellt werden, welche weiteren Codestellen und Programme von der Änderung betroffen sind. Um dies feststellen zu können, wurde ADS so erweitert, dass bei der Generierung für jedes Macro festgehalten wird, für welche Programme das Macro verwendet wird und welche

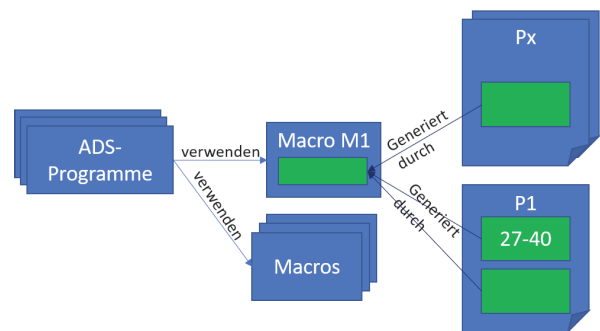


Figure 2: Woher stammt der Code? Und welche weiteren Programme sind betroffen?

Zeilen aus dem jeweiligen Programm durch das Macro erzeugt werden. Damit diese Aussage vollständig ist, muss die gesamte Anwendung generiert werden.

Dann gilt es festzustellen, welche Auswirkung eine Änderung des Macros auf alle anderen Codestellen und Programme hat. Wenn sichergestellt wurde, dass die Änderung auch für alle anderen Stellen unkritisch ist, kann das Macro geändert und die Anwendung neu generiert werden.

5 Generierte Anwendung modernisieren – Reengineering für Fortgeschrittene!

Auch generierte Anwendungen müssen modernisiert werden. Die Besonderheit liegt nun aber darin, dass es nicht ausreicht die Anwendung zu überarbeiten, sondern die Änderungen am Generator selbst vorzunehmen, damit dieser auch weiterhin eingesetzt werden kann.

Ist der Hersteller des Generatorsystems in den Modernisierungsprozess involviert, so besteht die Möglichkeit den Generator so anzupassen, dass alle notwendigen Informationen bei der Generierung erzeugt werden. Eine Modernisierung generierter Anwendungen verläuft in diesem Fall zwar anders als bei einer manuell codierten Anwendung, ist aber eine machbare Aufgabe.

Was aber, wenn der Hersteller nicht involviert ist? Ein Vorteil von Generatoren besteht darin, dass sie Code immer gleich erzeugt. Wie kann man das ausnutzen? Lassen sich Methoden wie Mustererkennung oder Clone-Detection so anpassen, dass man mit ihnen nicht nur gleiche oder ähnliche Codestellen in der generierten Anwendung aufdecken kann, sondern auch im Generator-Input? Welche anderen Reengineering-Methoden könnten angepasst wiederverwendet werden? Es scheint, die Modernisierung von generierten Anwendungen ist Reengineering für Fortgeschrittene!