



FAQ

KI-basierte Sprachtransformation von COBOL nach Java

1. Kann man ein Legacy-Softwaresystem mit nur einem KI-Tool migrieren?

Nein. In einem Migrationsprojekt kommt nicht nur ein Migrationstool zum Einsatz, sondern es sind mehrere notwendig. Gängige KI-Tools fokussieren auf die reine Programmcodekonvertierung. In einem Migrationsprojekt sind aber noch weitere Artefakte zu migrieren, z. B. JCL, Files, Datenbanken, Middleware und Masken. Das muss durch mehrere, spezielle Tools erledigt werden, um den Migrationsprozess zu automatisieren.

2. KI-Tools können doch aber bei der eigentlichen Programmkonvertierung, z. B. COBOL nach Java, Unterstützung bieten?

In COBOL z. B. existiert ein komplexes Datentypkonzept, für welches in Java keine 1:1-Abbildung existiert. Zusätzlich kommen in COBOL noch „Sprachspezialitäten“ wie etwa Überlagerungen von Variablen unterschiedlichen Datentyps und weitere dazu. Unsere Experimente mit verschiedenen KI-Tools haben gezeigt, dass bei KI-Tools im Wesentlichen COBOL-Datentypen in Java-Datentypen abgebildet werden. Die konvertierten Java-Programme sind zwar lauffähig, liefern aber falsche Ergebnisse. Die semantische Äquivalenz ist aber der wichtigste Faktor zwischen Alt- und Neusystem in einem Migrationsprojekt. Natürlich können diese Tools entsprechend trainiert werden. Um jedoch eine vollständige Abbildung von COBOL nach Java zu erreichen, ist ein großer Aufwand notwendig, der nach unseren Erkenntnissen dem einer Neuentwicklung nahekommt. Diese Abbildungen sind bei unseren Tools in Form einer Java-Bibliothek schon vorhanden, welche aktuell einen Abdeckungsgrad von 95–99 % erreicht. Um diesen Abdeckungsgrad in Migrationsprojekten zu erzielen, war eine jahr(zehnt)elange Entwicklungszeit notwendig. Speziell für die Sprachkonvertierung trainierte KI-Tools, z. B. COBOL nach Java, erreichen nach unseren Erfahrungen diesen Abdeckungsgrad nicht. Das bedingt dann manuelle Eingriffe und Erweiterungen in den konvertierten Java-Programmen. Ein großer manueller Aufwand und Fehleranfälligkeiten sind damit vorprogrammiert.

3. Kann ein KI-Tool prozeduralen COBOL-Code direkt in objektorientierten Java-Code migrieren? Ich habe sowas mal im Internet gelesen.

In den letzten Jahr(zehnt)en gab es eine Reihe wissenschaftlicher Arbeiten zum Thema: automatischer Umbau prozeduralen Codes in objektorientierten Code (unabhängig von einer Migration). Nach unserem Kenntnisstand führte keine dieser Arbeiten zu einem Ergebnis, welches zufriedenstellend praktisch nutzbar ist. Wenn solche Verfahren nicht existieren, kann es auch kein Werkzeug geben, welches solche Algorithmen automatisch implementiert. Das ist unabhängig davon, ob es sich um ein KI-Tool handelt oder nicht. Natürlich können zukünftige Weiterentwicklungen des migrierten Systems objektorientiert erfolgen, sodass im Laufe der Zeit zunehmend Objektorientiertheit dominiert, auch durch weitere Refaktorisierung des migrierten Programmsystems in der Wartung/Weiterentwicklung. Das bestätigt sich z. B. in Projekten, welche wir nach erfolgreicher Migration im Kundenauftrag warten und weiterentwickeln.



4. Welche Unterstützung können KI-Tools aktuell in einem Migrationsprojekt leisten?

Aktuell ergibt sich bei KI-Tools eine Reihe nützlicher Einsatzmöglichkeiten in einem Migrationsprojekt, z. B. bei der Testunterstützung, bei der Codekommentierung, bei der Refaktorisierung und der Programmdokumentation. Aber eben nur als Assistenten. Beim IBM watsonx Code Assistant for Z ist diese Funktionalität schon im Namen hinterlegt.

Wenn der Kunde in überschaubarer Projektzeit und mit überschaubarem Budget einen Wechsel der Programmiersprache und/oder einen Plattformwechsel wünscht, dann ist aktuell die toolgestützte, compilerbasierte Software-Migration das Mittel der Wahl. KI-Tools können dabei wertvolle, assistenzbasierte Unterstützung leisten.