

## Konvertierung von Host-Masken in moderne Benutzeroberflächen

Auszug aus "Methoden und Werkzeuge für die Software Migration" (Artikel zum 10. Workshop "Software Reengineering" am 05.-07. Mai 2008 in Bad Honnef, erschienen in: *10th Workshop Software Reengineering, 5-7 May 2008, Bad Honnef. Lecture Notes in Informatics (LNI) Proceedings, Series of the Gesellschaft für Informatik (GI), Volume P-126, 2008*)

Die Konvertierung ASCII-orientierter Host-Masken in browserbasierte Benutzeroberflächen soll am Beispiel von SCREEN COBOL expliziert werden. Mit diesem COBOL-Dialekt werden Masken z.B. in HP NonStop-Systemen (Tandem) entwickelt. Die Konvertierungstechnologie läßt sich jedoch vom konkreten Basissystem abstrahieren und verallgemeinern. Abbildung 3 dokumentiert das Basissystem:

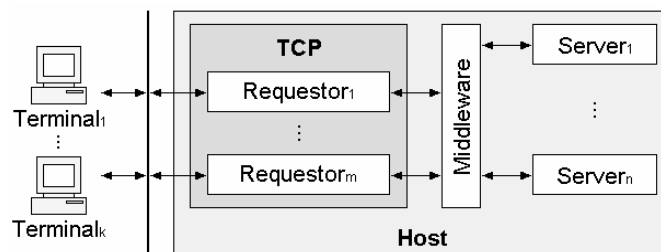


Abbildung 1: HP NonStop-Basissystem

Die in SCREEN COBOL geschriebenen Maskenprogramme (sogenannte Requestoren) laufen unter Steuerung eines Terminal-Control-Prozesses (TCP) und zeigen an den über eine Terminalemulation angeschlossenen Terminals die Maskeninhalte an. Erfolgen in den Masken Eingaben, werden diese an den Requestor übertragen. Unter Einbindung einer Middleware wird daraufhin dem Server (ein transaktionsorientiertes Programm) eine Message gesendet. Der Server realisiert die Verarbeitung und sendet eine Antwortmessage. Diese wird vom Requestor verarbeitet und als Ergebnis werden neue Daten in der aktuellen Maske angezeigt oder eine andere Maske aufgeschaltet etc. SCREEN COBOL-Programme beinhalten zwei für den Konvertierungsprozeß wichtige Informationen:

- Die Anordnung der Ein- und Ausgabefelder in der Maske (das "Layout") einschließlich ihrer Eigenschaften.
- Die COBOL-Strukturen, in welchen die Ein- und Ausgabeinformationen gespeichert werden.

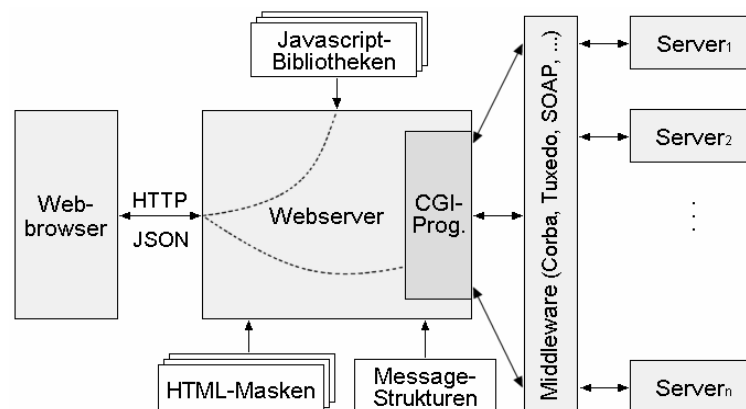


Abbildung 2: Technologie der Maskenkonzertierung

Für die Konvertierung sind diese Informationen wesentlich, sie werden automatisiert aus den SCREEN COBOL-Programmen extrahiert. Abbildung 4 verdeutlicht das: Es existiert ein Werkzeug *SC-Konverter*, Hauptbestandteil ist ein front-end für *SCREEN COBOL-Programme*. SC-Konverter liefert zwei Ergebnisfiles:

- *Message-Strukturen* beinhalten in aufbereiteter Form die o.g. COBOL-Strukturen.
- *Screen-Informationen* beinhalten Informationen zum Maskenlayout. Daraus generiert ein Werkzeug *ScreenConv* HTML-Code für die Masken im Zielsystem (*HTML-Masken*).

Message-Strukturen und Screen-Informationen werden im Zielsystem genutzt.

In einem konkreten Migrationsprojekt existierten für die Gestaltung der Benutzerkommunikation im Zielsystem die folgenden Nutzerforderungen:

- Der Maskenaufbau sollte erhalten bleiben. Der Nutzer motivierte das mit einer kurzen Einarbeitungszeit.
- Die Lauffähigkeit im Webbrowser im Zielsystem sollte ohne zusätzliche Installation auf dem Clienten gegeben sein. Hier spielten reduzierter Administrationsaufwand und Einsparung von Lizenzkosten eine Rolle.

In Abbildung 5 wird die Zielsystemarchitektur des Migrationsprojektes dokumentiert.

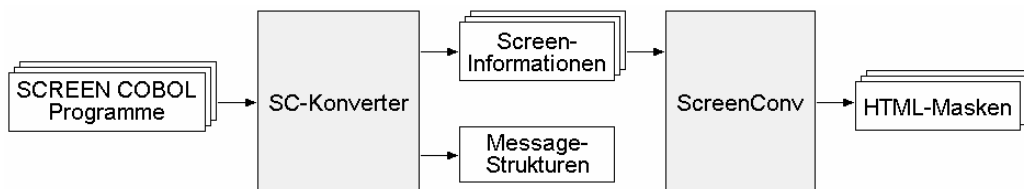


Abbildung 3: Architektur des Zielsystems

Für die Kommunikation zwischen Webbrowser und Webserver wird AJAX genutzt. Das verhindert unnötige Reloads von kompletten Seiten und den Eintrag in die History des Browsers, die im Basissystem auch nicht vorhanden ist.

Die Kommunikation stellt sich im Zielsystem dann wie folgt dar:

Eine Maske wird im Browser angezeigt. Ihr HTML-Code wurde auf Basis der SCREEN COBOL-Programme des Basissystems (teil)automatisch generiert und steht als HTML-File zur Verfügung. Die Maske nutzt eine Javascript-Bibliothek für die Kommunikation mit dem Server und eine maskenspezifische Javascript-Bibliothek für Aktionen wie das Auslesen von Eingaben, das Setzen von neuen Werten in der Maske sowie die Steuerung. Beide Bibliotheken sind in Zielsystemen obiger Architektur wiederholt einsetzbar. Werden Eingaben in der Maske getätigt und mit einer dem Basissystem entlehnten "Datenfreigabe" (ENTER-Taste) beendet, so werden die Maskendaten über einen AJAX-HTTP-Request an den Webserver gesendet. Für diese Kommunikation wird JavaScript Object Notation (JSON) als Alternative zu XML verwendet. Der Webserver übergibt die Daten an ein CGI-Programm, welches die Message vom JSON-Format in eine COBOL-Message konvertiert. Eine COBOL-Message ist vereinfacht eine mit Daten gefüllte COBOL-Struktur. Die für das Füllen der COBOL-Struktur notwendigen Informationen (Offset, Länge, Typ der einzelnen Datenfelder) stehen in Message-Strukturen zur Verfügung. Sie wurden im Konvertierungsprozeß aus den SCREEN COBOL-Programmen gewonnen. Die mit Inhalten gefüllte COBOL-Struktur wird über eine Middleware (CORBA, TUXEDO,...) an den COBOL-Server gesendet. In der Referenzimplementierung wird anstelle einer kommerziellen Middleware das handliche SOAP-Protokoll verwendet. Der Server verarbeitet die Message, erzeugt eine Antwortstruktur und sendet diese zurück zum CGI-Programm. Dieses erzeugt aus den Daten wieder eine JSON-Struktur und schickt sie mit HTTP-Request an den Browser. Über Javascript-Funktionen werden die entsprechenden Daten zur Anzeige gebracht. Soll auf Grund der Serverantwort eine neue Maske (HTML-Seite) angezeigt werden, wird diese vorher durch einen weiteren HTTP-Request geladen und angezeigt.

Erfahrungen mit dieser Zielarchitektur besagen, daß die Konvertierung von SCREEN COBOL nach HTML und Javascript möglich ist und eine Reihe von Vorteilen (Unabhängigkeit von proprietären, kostenpflichtigen Maskentools bzw. Emulatoren, Einsatz von Standardwerkzeugen und -techniken, ...) bietet.