

Die Rettung der "Alten Welt"

Anwendungs-Migrationen und -Portierungen von Legacy-Systemen in die „offene Welt“

Teil II - Anwendungs-Portierung und -Modernisierung Proprietäre Systeme mit herstellereigenen Betriebssystemen

Verfasser: Andreas Maier
OrgaBIT
Organisation • Beratung • IT

Datum: 10.09.2008

Nachdem wir im Teil I die Begriffe Migration und Portierung erläuterten und die möglichen Migrationsstrategien vorstellten, soll hier nun näher auf den von uns präferierten Migrationsweg einer Software-Portierung mit einhergehender Modernisierung bzw. teilweisem Reengineering der Anwendung eingegangen werden.

Nochmals zur Erinnerung: eine Portierung ist die physische Übernahme von Anwendungsprogrammen und -daten auf eine neue Betriebssystem-Umgebung und die Konvertierung des existierenden Sourcecodes und der Nutzdaten damit diese in der neuen Umgebung fehlerfrei genutzt werden können.

Eine Portierung bietet sich grundsätzlich dann an, wenn die betreffende Anwendung auf die Prozesse des Unternehmens optimal zugeschnitten ist, die Anwendung als geschäftskritisch eingestuft werden kann und natürlich auch durch Individual-Programmierung entstanden ist.

Natürlich stellt sich immer die Frage, auf welche Hardware-/Betriebssystem-Kombination heute normalerweise eine Anwendungs-Migration geplant und durchgeführt wird: aufgrund der Offenheit, der freien Verfügbarkeit, der Unterstützung durch alle großen IT-Hersteller und damit des weltweiten Supports und nicht zuletzt aufgrund der Verfügbarkeit einer immens großen Anzahl jeglicher Art von Middleware, Werkzeugen und Bibliotheken wird als Zielplattform für eine Anwendungsportierung heute typischerweise immer ein Linux-basierendes System gewählt. Bei einigen Portierungen kommen auch die vom jeweiligen Hersteller angebotenen Systeme mit einem der Unix-Derivate zum Einsatz, weil diese Systeme teilweise schon Erweiterungen beinhalten, welche den Portierungsvorgang einfacher machen können.

Da für die in Teil I erwähnten betroffenen Typen von Systemarchitekturen sehr unterschiedliche Portierungsansätze gelten und die Fragestellungen und Vorgehensweisen während einer Portierung sehr unterschiedlich ausfallen, werden in diesem Teil nur die Klasse der „proprietären Systeme mit herstellerspezifischem Betriebssystem“ abgehandelt. In Teil III gehen wir detailliert auf die Vorgehensweisen bei Anwendungsportierungen von „RISC-Systemen mit Unix-Derivat“ ein.

Proprietäre Hardware/Architektur mit herstellerspezifischem Betriebssystem und proprietärer Middleware

Ein paar Beispiele für diese Klasse von Systemen sind

Hersteller / Rechnerfamilie	Betriebssystem + Middleware
HP HP1000	O/S: RTE / RTE-A (RealTime Executive) Datenbank: IMAGE/1000 Maskensystem: FORMS/1000 Networking: DS und TCP/IP Shell/CLI: CI Prog.sprachen: Fortran, Pascal, C, Macro-Assembler System-API: umfangreiche System-Bibliotheken für Dateizugriffe und Kernelroutinen

Hersteller / Rechnerfamilie	Betriebssystem + Middleware	
HP HP3000 / e3000	O/S: Datenbank: Maskensystem: Networking: Shell/CL: Prog.sprachen: System-API:	MPE / MPE/iX (MultiProgramming Executive) TurboIMAGE/XL, Allbase/SQL VPLUS DS und TCP/IP CI Cobol, RPG, Fortran, Pascal, Basic, SPL weniger umfangreiche System-Bibliotheken für Dateizugriffe und Kernelroutinen
HP/DEC VAX/Alpha	O/S: Datenbank: Maskensystem: Networking: Shell/CLI: Prog.sprachen: System-API:	VMS/OpenVMS (Virtual Memory System) RDB, RMS DECforms DECnet / TCP/IP DCL Cobol, Fortran, Pascal, C/C++, Basic sehr umfangreiche System-Bibliotheken für Dateizugriffe und Kernelroutinen
IBM AS/400 / System i	O/S: Datenbank: Maskensystem: Networking: Shell/CLI: Prog.sprachen: System-API:	OS/400, i5/OS, IBM i (Operating System 400) Integrated DBMS, DB2, VSAM Integriert Token Ring / TCP/IP CL RPG, Assembler, C/C++, Pascal, Cobol, SQL, Basic, PL/I sehr umfangreiche System-Bibliotheken für Dateizugriffe und Kernelroutinen
Diese Liste erhebt keinerlei Anspruch auf Vollständigkeit und soll nur beispielhaft einige Systeme mit dazugehörigem Betriebssystem und der verfügbaren Middleware aufzeigen		

Wie aus der beispielhaften Liste ersichtlich, zeichnet die Systeme allesamt aus, dass Sie in jeder Hinsicht proprietär sind, das heißt, zu den so genannten „geschlossenen Systemen“ (closed systems) gehören. Betriebssystem, Middleware, Entwicklungswerkzeuge und die mitgelieferten Systemwerkzeuge sind herstellerspezifisch und nur auf diesen Plattformen verfügbar. Die verfügbare Dokumentation sagt nur etwas über die Funktionalitäten und die vorhandenen Schnittstellen aus, gibt aber keine Details Auskunft über interne Datenstrukturen, Dateiformate und Implementierung.

All das kann eine Anwendungsportierung von diesen Plattformen sehr schwierig und umfangreich gestalten, weshalb hier eine Portierung nur dann Sinn macht, wenn entsprechende Konvertierungs- und Portierungswerkzeuge zur Verfügung stehen.

Portierungswerkzeuge und Portierungsprojekte

Im Allgemeinen kann gesagt werden, dass die verfügbaren Portierungswerkzeuge und die Portierungsprojekte immer nach dem folgenden Schema arbeiten bzw. ablaufen:

1. Über eine Analysetool wird die zu portierende Code-Basis untersucht, mit dem Ziel, eventuell vorhandene Inkompatibilitäten aufzuzeigen und eine Aufwandsabschätzung durchführen zu können.
2. Auf der neuen Plattform wird eine passende Verzeichnis-Struktur und Entwicklungsumgebung aufgebaut
3. Der zu portierende Sourcecode wird automatisch so konvertiert, dass er mit den Compilern auf der neuen Plattform fehlerfrei übersetzt werden kann.
4. Die auf dem bisherigen System verfügbare Middleware und System-Bibliotheken werden auf der neuen Plattform nachgebildet/emuliert oder über Konvertierung durch ein Standardprodukt ersetzt (Datenbanken, Maskensysteme, etc.).
5. Anwendungsteile, welche in einer Programmiersprache entwickelt wurden für die es auf der neuen Plattform keinen Compiler gibt (z.B. Assembler, PL/I) werden entweder in eine andere Programmiersprache konvertiert oder neu programmiert.
6. Sofern sinnvoll und möglich, Nachbildung der Scriptsprache des Quellsystems auf dem Zielsystem oder alternativ Neu-Erstellung der Scripts.
7. Vorhandene Datenbestände müssen gegebenenfalls konvertiert werden (EBCDIC ↔ ASCII, Little Endian ↔ Big Endian, Floating Point Darstellung) und können dann übernommen werden.
8. Wo möglich und sinnvoll, Modernisierung der Anwendung im Rahmen der Portierung (z.B. Graphical User Interface / Web-Interface; Austausch einer ISAM-orientierten Datenhaltung durch ein relationales Datenbanksystem, etc.).
9. Während der Funktions- und Modultests (Unit-Tests), gegebenenfalls Erweiterung oder Anpassung der Konvertierungswerkzeuge und der Emulations-Bibliotheken.

Als Beispiel soll hier näher auf das Portierungswerkzeug **RTE-UX** der Firma OrgaBIT eingegangen werden: RTE-UX dient dazu, Portierungen von der HP1000-Plattform mit dem RTE-A Betriebssystem in Richtung Unix und Linux möglichst einfach, schnell und damit kostengünstig zu gestalten.

Hewlett-Packard HP1000

Die von Hewlett-Packard (HP) in den siebziger Jahren vorgestellte HP1000-Rechnerfamilie (ursprünglich HP21000) war ursprünglich als Steuerungsrechner für die von HP angebotenen Messgeräte gedacht. Ausgestattet mit dem Multiuser/Multitasking-Betriebssystem RTE (Real-Time Executive) erweiterte sich das Anwendungsspektrum nach und nach in Richtung allgemeiner Steuerungs- und Überwachungsaufgaben von jeglicher Art technischer Anlagen. Mit der laufenden Weiterentwicklung der Architektur

wurde auch das Betriebssystem modernisiert und erweitert: ein hierarchisches Datenbanksystem (IMAGE/1000) und ein Maskensystem (FORMS/1000) wurden in das Produktportfolio mit aufgenommen und umfangreiche Vernetzungsmöglichkeiten (LAN/WAN) implementiert. Die Programmierung erfolgte in den Sprachen Macro-Assembler, Fortran, Pascal und Basic, von Drittanbietern gab es einen C-Compiler. Die Schnittstelle zum Betriebssystem gestaltete sich in Form einiger umfangreicher Bibliotheken (z.B. FMP / File Management Package, EXEC / I/O, IPC, Kernelschnittstelle). Ermöglicht durch den erweiterten Funktionsumfang, wurden auf dem HP1000-System sogar in vielen Firmen individuell programmierte kommerziell orientierten Anwendungen eingesetzt.

Die letzten HP1000 Produkte wurden von Hewlett-Packard am 1. November 2000 vom Markt genommen, das Ende des Supports für die CPUs vom Typ A400, A600, A900 und A990 erfolgte im Jahr 2005.

Viele der Kunden, welche auf der HP1000 große Individual-Anwendungen unter Nutzung der verfügbaren Middleware entwickelten, haben das Bestreben, diese Anwendungen zu retten und auf die neue – typischerweise Unix-basierende – Plattform zu übernehmen.

OrgaBIT RTE-UX

RTE-UX wurde dahingehend entwickelt und optimiert, Anwendungs-Portierungen von der HP1000-Plattform in Richtung Unix und Linux möglichst ohne große Änderungen am ursprünglichen Programmcode möglich zu machen. Da die meisten großen Anwendungen auf der HP1000 in Fortran entwickelt wurden, legt RTE-UX ein besonderes Augenmerk auf diese Programmiersprache.

Der wesentliche Funktionsumfang von RTE-UX ist aus folgender Liste ersichtlich:

- » Sourcecode Konverter (HP1000/Fortran to Fortran 90)
- » Emulation der meisten RTE-A System Intrinsic - EXEC, FMP, SYSLIB, etc.
- » Unterstützung von LU's (Logical Units), RTE CLASS I/O (Interprocess Communication), Resource Numbers, Shared Memory, etc.
- » Emulation des FMP-Subsystems (verschiedene Dateitypen, locking)
- » API zum Unix Print Spooling System
- » Unterstützung für das IMAGE/1000 DBMS
- » Unterstützung für das FORMS/1000 Maskensystem
- » BSD IPC Sockets/1000 API zur Unix Sockets Library
- » API zur Unix syslog Utility
- » Subset der CI-Kommandos speziell für Prozesskontrolle

Neben der reinen Portierung einer HP1000-Anwendung unterstützt RTE-UX zugleich auch noch diverse Modernisierungsschritte. So kann z.B. das zeichen- bzw. feldorientierte Maskensystem FORMS/1000 ohne Änderungen am Sourcecode durch ein Graphical User Interface ersetzt werden (Stichwort GUIifizierung) oder auch das hierarchische Datenbanksystem IMAGE/1000 durch ein modernes DBMS mit SQL-Schnittstelle.

Fallbeispiel für eine Portierung mit RTE-UX

Eine der größten nordamerikanischen Eisenbahngesellschaften transportiert per Güterzügen jegliche Art von Waren, wie z.B. Petroleum, Getreide, Kohle, Metalle und Mineralien, auf dem nordamerikanischen Kontinent vom Atlantik bis zum Pazifik in Ost-West-Richtung und von Kanada bis zum Golf von Mexiko in Nord-Süd-Richtung.

Die zu befördernden Waren, über ein Produktionsplanungssystem optimal auf Güterwägen verteilt und zu Güterzügen kombiniert, werden in sogenannten "Classification Yards" oder auch "Hump Yards" (Rangierbahnhöfe) über einen Ablaufberg zusammengestellt. Ein Ablaufberg besteht immer aus einer oder mehrerer Eingangsspuren und bis zu 100 Ausgangsspuren. Dabei rollen die einzelnen Güterwägen, über die Eingangsspuren auf einen kleinen Hügel geschoben, den Abhang hinunter und werden über das "Hump Process Control System" durch Umlegen von Weichen auf die richtige Ausgangsspur geleitet. Das Unternehmen betreibt derzeit 4 dieser Classification Yards in Kanada und den USA.

Genau diese Anwendung, das Hump Process Control System (HPCS), wurde ursprünglich mit einem Aufwand von ca. 100 Mannjahren hauptsächlich in der Programmiersprache Fortran/77 auf der HP1000-Plattform entwickelt. Da die HP1000-Plattform seit einigen Jahren von HP nicht mehr unterstützt wird, gab es im Jahr 2005 eine strategische Entscheidung des Unternehmens, die installierten Rechner durch eine neue, moderne Rechnergeneration zu ersetzen.

Die dann durchgeführten Untersuchungen bzgl. möglicher Migrationsstrategien (Neuprogrammierung, Beschaffung einer Standardanwendung, Software-Portierung) haben ergeben, dass der Weg einer Software-Portierung unter Nutzung des von OrgaBIT entwickelten Portierungswerkzeuges RTE-UX den schnellsten und kostengünstigsten Erfolg bringen würde. Neben der Erhaltung der Investitionen in die vorhandene Anwendung konnten so auch die portierten Programme mit notwendigen technischen Erweiterungen ausgestattet werden und das KnowHow der betroffenen Mitarbeiter weiter genutzt werden.

So konnte das Hump Process Control System auf Basis von RTE-UX und mit Hilfe von OrgaBIT binnen ca. 6 Monaten auf die HP9000/HP-UX-Plattform unter Nutzung des HP Fortran/90-Compilers portiert werden und die Classification Yards nach und nach auf die portierte Anwendung migriert werden.

Durch die Portierung der bestehenden Anwendung auf ein Unix-System konnten im Laufe des Projektes auch viele Neuerungen und Erweiterungen in die Software eingearbeitet werden. So wurde z.B. eine Fortran-Schnittstelle in die IBM MQ-Series API geschaffen, um mit dem Produktionsplanungssystem des Unternehmens Produktions- und Planungsdaten austauschen zu können.

© ® Alle Markennamen, Warenzeichen und eingetragenen Warenzeichen der in diesem Dokument aufgeführten und erwähnten Produkte und Dienstleistungen sind Eigentum Ihrer rechtmäßigen Eigentümer und dienen hier nur der Beschreibung. Das Kopieren von Daten (jeglicher Art) aus diesem Dokument ist ohne schriftliche Zustimmung von OrgaBIT grundsätzlich untersagt.
