

Beziehungsprobleme – 50 Jahre Portabilität

Die Portabilität ist in der Midlife-Crisis. Es wird immer schwieriger, Anwendungen in verschiedensten Umgebungen lauffähig zu gestalten, respektive Daten in verschiedensten Umgebungen lesbar und auswertbar zu machen.



« Die Entwicklung der letzten Jahre hat die vergangenen Fortschritte in Sachen Portabilität vollständig zunichte gemacht. »

Daniel Liebhart ist Dozent für Informatik an der ZHAW (Zürcher Hochschule für Angewandte Wissenschaften), Experte für Enterprise-Architekturen und CTO Ambassador bei Swisscom. Er ist Autor und Co-Autor verschiedener Fachbücher.

Vor 50 Jahren hat es der IT-Pionier Alan J. Perlis zum ersten Mal an einer NATO-Konferenz in Garmisch-Patenkirchen formuliert: «Portabilität ist die Eigenschaft eines Systems, die es erlaubt, dieses von einer Umgebung in eine andere Umgebung zu übertragen.» Die Definition ist nach wie vor aktuell, wie ein Blick in das IEEE-Glossar für Software-Engineering zeigt. Im weitesten Sinne ist Portabilität ein Mass, das den Aufwand für die Übertragung eines Systems von einer Umgebung in eine andere ins Verhältnis zum Aufwand für eine Neuentwicklung stellt. Ist der Neubau aufwändiger als die Anpassung, so gilt das System als portabel. Lange Zeit haben wir uns mit zwei Mitteln beholfen, um die Portabilität zu vereinfachen: Standards respektive standardisierte Schnittstellen oder die zusätzliche Abstraktion über Schichten.

Beziehungskitt mittels Standards

Wegbereiter für Standards als Beziehungskitt zwischen unterschiedlichen Umgebungen war der 1980 von der IEEE veröffentlichte POSIX-Standard. «POS» steht gemäss Richard Stallmann, der diesen mitgestaltete, für «Portable Operating System», und «IX» ist die Anlehnung an UNIX. Ziel des Standards ist es, die Portabilität von Anwendungen zwischen verschiedensten Betriebssystemen zu vereinfachen. Das Mittel dazu ist denkbar einfach. Funktionalität wird einheitlich geschnitten und über genau definierte Schnittstellen zur Verfügung gestellt. Das vereinfacht deren Nutzung. Diese Praxis wird heute in nahezu jeder Industrie verwendet, um den Aufwand für die Befriedigung des rasant steigenden Softwarebedarfs in Grenzen zu halten. Der ursprüngliche POSIX-Standard hat sich aus gutem Grund gehalten und ist heute als internationale Norm

SO/IEC/IEEE 9945 verfügbar. Dieser Standard wird von jedem Betriebssystem, das diesen Namen verdient, unterstützt.

Schichten durch Schichten

Dem Siegeszug der beiden heute wichtigsten Programmiersprachen (Java und C#) liegt eine weitere einfache Idee zugrunde. Verschiedene Umgebungen werden durch die Einführung einer zusätzlichen Schicht vereinheitlicht. Diese verwendet eine virtuelle Maschine, die es ermöglicht, dass sich unterschiedlichste Systeme auf dieselbe Art und Weise verhalten. Ob Grossrechner, Industrieroboter, Fahrzeug oder Smartphone – überall können Anwendungen ausgeführt und Daten gelesen werden.

Die Problematik

Die Entwicklung der letzten Jahre hat jedoch die vergangenen Fortschritte in Sachen Portabilität vollständig zunichte gemacht. Allein schon die Art und Weise, wie wir heute Cloud- oder Smartphone-Anwendungen bauen, zeigt, wohin die Reise geht – weg von der Portabilität hin zu den «nativen» Lösungen. So ist es weitaus einfacher, je eine App-Variante für ein iPhone und eine für ein auf Android basierendes Phone zu bauen, als eine portable Lösung umzusetzen. Und niemand denkt auch nur im Traum daran, ein grösseres portables System für die drei grossen Cloud-Plattformen Azure, Amazon und Google zu gestalten. Viel zu unterschiedlich sind deren innerer Aufbau und Funktionsaufteilung. Im Bereich der Portabilität von Daten sieht es nicht besser aus. Die Explosion der Formate macht den Zugriff, die Aufbereitung und den Schutz von Daten immer schwieriger. Es wird Zeit für etwas Beziehungspflege, um das zu ändern.

Artikel online

auf www.netzwoche.ch
Webcode DPF8_87345