

Viele Wege führen zu modernen IBM i-Anwendungen – Teil 1

Modernisierung à la LANSA

Wer heute vor der Aufgabe steht, seine bestehenden „Green Screen“-Anwendungen zu modernisieren, dem stehen mehrere Wege offen. Ein Ausgangsproblem bei der Modernisierung stellt sich in den allermeisten Fällen: Die Menüstruktur der 5250-basierten Anwendung bleibt bestehen. Doch die Anwender verlangen nach einer komfortableren Lösung, sprich die Modernisierung bezieht sich auch darauf, zusätzliche Funktionalität in die Anwendung einzufügen. Hier bieten sich mit den Entwicklungswerkzeugen von LANSA zukunftsichere Lösungswege an.

Jedes Softwaremodernisierungsprojekt bedarf einer möglichst präzisen Zielvorgabe. Dabei sind einige Punkte zu berücksichtigen: Wichtig ist vor allem eine passgenaue Analysephase, zudem sollte die Zielerreichung messbar und mit überschaubarem Aufwand realisierbar sein. Betrachtet man die Modernisierung von IBM i-Anwendungen aus einer Art Vogelperspektive, kommen drei Vorgehensweisen in Betracht:

- der Einsatz einer neuen Programmiersprache
- die Verwendung von speziellen Applikationen
- der Einsatz von leistungsfähigen Entwicklungstools

Wer den ersten Weg ins Auge fasst, der sieht sich häufig mit einer hohen Komplexität konfrontiert: Denn oftmals reicht eine zusätzliche Programmiersprache nicht aus und das Erlernen neuer Sprachen geht mit großem Zeitaufwand einher. Zudem gibt es in der Regel keine Unterstützung im eigenen Haus, denn das benötigte (neue) Know-how ist nicht sehr ausgeprägt. Ein weiteres Problemfeld ist das Thema Wartung der Software. Auch hier zeichnet sich ein höherer Aufwand ab, denn man kennt noch zu wenig unterstützende Werkzeuge für die neue Programmierumgebung.

Bei der zweiten Vorgehensweise, dem Rückgriff auf eine passende Appli-

kation, gilt es ebenfalls einige Klippen zu umschiffen. Das beginnt bereits bei der Suche nach der Applikation, die den Vorgaben am besten gerecht wird. Denn sollten sich im Laufe des Projekts die Anforderungen ändern, steht man vor einer erneuten Suchphase – und die erfordert wieder Zeit. Es geht bei Modernisierungsprojekten folglich auch darum, alle verwendeten Schnittstellen zu prüfen. Nur so lässt sich ausschließen, dass sich die eingesetzten Applikationen zu einem späteren Zeitpunkt nicht mehr verständigen können.

Die dritte – und optimale – Möglichkeit der Modernisierung ergibt sich durch den Einsatz von leistungsfähigen Entwicklungstools. Sie bilden im Gegensatz zum Einsatz verschiedener Applikationen ein in sich geschlossenes System – und zwar für jede Aufgabe, so verschieden sie auch sein mag. Der Hersteller des Tools kümmert sich um die Weiterentwicklung: Somit wird das Tool immer wieder auf den neuesten Stand gebracht. Das heißt, Anforderungen, die sich durch die fortschreitende Technik ergeben, können vernachlässigt werden – mit dem Einspielen eines Updates ist zumindest theoretisch alles erledigt. Auch das Argument der Effizienz spricht für den Einsatz eines Tool-Sets. Verglichen mit dem Rückgriff auf eine neue Programmiersprache ergibt sich ein dreimal so

hoher Effizienzwert. Für den „Modernisierer“ bleiben nur noch wenige Aufgaben: Lediglich das Front-End ist zu gestalten und die Datenbankzugriffe sind noch zu definieren.

Mit den Produkten von LANSA steht ein solches Tool-Set zur Verfügung: Durch den integrativen und modularen Aufbau gibt es für alle Wege des Modernisierungsprozesses die richtigen Werkzeuge und Lösungen.

Einsatz von LANSA aXes

Meist geht es im ersten Modernisierungsansatz darum, eine 5250-Anwendung „ins Web zu bringen“. Zudem sollte alles über eine moderne, „mausorientierte“ Oberfläche, mit benutzerfreundlichen Bedienelementen wie Dropdown-Menüs oder Push-Buttons realisiert werden. Im Zuge eines Modernisierungsprojekts bietet es sich natürlich auch oftmals an, den Funktionsumfang der Lösung an neuere Anforderungen anzupassen. So kann es zum Beispiel nötig sein, aus der 5250-Anwendung heraus über einen Link direkt auf Dokumente im Archiv zuzugreifen oder im Kundengespräch auf PDF-Dokumente mit Produktbeschreibungen zu verzweigen.

Diese Funktionserweiterungen sollten allerdings ohne Änderungen im RPG- oder im Cobol-Sourcecode möglich sein. Auch eine Anpassung des

DDS-Quellcodes ist in diesem Kontext meist nicht erwünscht. Des Weiteren soll die neue Software dann nicht nur auf einem „normalen“ Bildschirm eine gute Figur machen, sondern auch auf einem Mobilgerät wie einem Tablet oder einem Smartphone.

Allerdings darf man für diese Art der Modernisierungsaufgabe eine wesentliche Voraussetzung nicht außer Acht lassen: Im Unternehmen sind die Anwender mit der grundlegenden

ten. Nach dem Modernisieren dieser Masken steht alles sofort zur Verfügung – per Web – und ohne Eingriffe in den Sourcecode.

Für die Modernisierung selbst gibt es bei aXes diverse Designelemente. Sie werden als Extensions bezeichnet. Mit diesen Objekten lassen sich die Masken neu gestalten. So kann man zum Beispiel Bilder oder Bedienelemente einbauen. Programmierer sind zudem in der Lage, eigene Extensions

Anforderung könnte sein, Informationen aus anderen Systemen via Webservices abzuholen und in die eigenen Abläufe einzufügen. Derartige Aufgaben lassen sich mit Visual LANSA einfach realisieren.

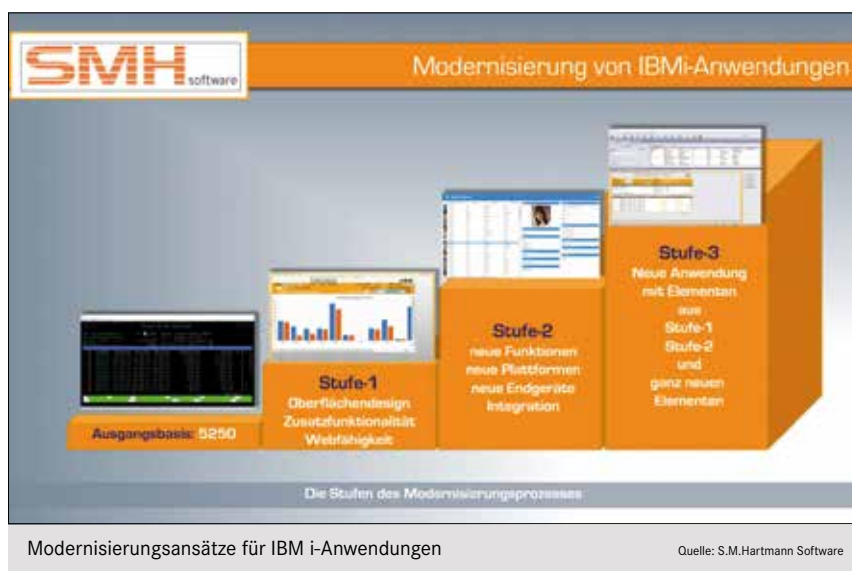
Heute weiß der Programmierer oft nicht, mit welchen Endgeräten letztendlich ein Benutzer auf die Applikation zugreift: Smartphones, Tablets und andere Mobilgeräte dürfen dabei nicht ausgenommen sein. Mit Visual LANSA erstellte Anwendungen präsentieren die Informationen auf jedem Endgerät, und zwar passend zum Formfaktor. Auch die Kommunikation verschiedener Datenbanken untereinander – wie etwa der Datenbank auf der IBM i mit einem SQL Server – ist in manchen Anwendungsfällen nötig. Da stößt reine RPG-Programmierung an ihre Grenzen.

Visual LANSA ermöglicht es, die IBM i-Datenbanken und die entsprechenden Dateien in dieses Entwicklungstool zu importieren. Dadurch ist der einfache Zugriff auf die Dateifelder gegeben, ebenso der Zugriff auf die logischen Dateien inklusive aller Schlüsselfelder. Es wird sozusagen eine identische Programmierenebene auf eine IBM i-Datei gelegt, die sich dann auf PC- oder Web-Ebene ansprechen lässt.

Daraus ergibt sich langfristig gesehen ein weiterer Vorteil: Wenn die Entscheidung fallen sollte, dass die AS/400 ausgetauscht werden soll, steht die ursprüngliche IBM i-Datenbank als SQL-Datenbank in den anderen Welten zur Verfügung. Und da Visual LANSA ein Tool ist, mit dem Programme für IBM i, PC, Client-Server und Web-Browser erstellt werden können, leistet Ihnen dieses Entwicklungswerkzeug auch „in einer anderen Welt“ beste Dienste.

Der zweite Teil des Beitrags in der nächsten Ausgabe skizziert den dritten Ansatz, den Königsweg der Modernisierung, der mit Hilfe von LANSA RAMP angegangen werden kann.

**Siegfried Hartmann und
Jörg Hamacher ■**



IBM i-Anwendung und der Datenbank im Großen und Ganzen noch zufrieden, und es besteht noch Zugriff auf den Programmierer dieser Software, so dass die Anwendung immer noch selbst gepflegt werden kann.

Um diese Aufgabenstellung zu erfüllen, muss ein „Modernisierer“ im ersten Ansatz nur ein passendes Tool – wie etwa LANSA aXes – installieren, und sofort ist der Zugriff auf die Anwendung auf der IBM i via Browser möglich. Dabei gilt: Im Browser läuft dieselbe Anwendung wie auf der IBM i. Änderungen am Sourcecode sind nicht nötig, und es muss auch kein Open Access Handler oder Ähnliches dazwischengeschaltet werden. Danach kann man die Maske benutzerfreundlicher und intuitiver gestalten

zu schreiben und sie zum Einsatz zu bringen.

Modernisieren mit Visual LANSA

In komplexeren Modernisierungsprojekten ist eine Integration auf verschiedenen Ebenen gefragt – das lässt sich über ein Werkzeug wie „Visual LANSA“ erledigen. So ist es zum Beispiel sinnvoll, verschiedene Systeme miteinander zu koppeln: Beispielsweise möchte man aus dem Warenwirtschaftssystem E-Mails an die Kunden versenden und dabei auch gleich PDF-Dateien anhängen, sei es im ZUGFeRD-Format oder Ähnlichem. Oder die Aufgabenstellung lautet, Rechnungen bzw. Bestellungen auszutauschen und sie als XML-Dateien anschließend auf einem externen FTP-Server abzuliegen. Eine andere