

Ein Besuch in der „alten“ Videothek



Wie die Zeit vergeht, sieht man an einem Szenario, dass MD Consulting jahrelang im Seminar „Team Developer Client-Server“ verwendet hat. In vier Tagen wurde am Beispiel einer Videothek die objektorientierte Programmierung mit SQLWindows unterrichtet. Videotheken sind aufgrund der übermächtigen Konkurrenz im Internet mehr oder weniger ausgestorben. Ich habe mir bei der Durchsicht der Unterlagen die Frage gestellt, ob man mit der aktuellen Version von Team Developer in der gleichen Zeit mit den gleichen Themen das gleiche Ergebnis erreichen würde.

Einleitung

In den letzten Jahren haben viele Kunden das Grundlagenseminar „Team Developer Client Server“ besucht und dabei das Vergnügen gehabt, sich von dem (mittlerweile pensionierten) Dozenten Alois Fuchs in die Tiefen und Fallstricke der objektorientierten Programmierung mit SQLWindows einführen zu lassen. Vor ein paar Tagen habe ich mir den Spaß gegönnt, mir die Unterlagen und Beispiele der seinerzeit mit Team Developer 4.x erstellten Anwendung anzuschauen. Im Ergebnis finde ich, dass nach wie vor der Aufbau und die Feingliederung dieses viertägigen Seminars Sinn machen.



Der auf vier Tage angelegte Einführungskurs sah damals folgende Themen vor:

- Einführung, Navigation im Outliner, Prinzipien der Programmierung, einfache Klassen
- DB Anbindung, Klassendesign advanced, dyn. Sql
- Klassen benutzen, Anwender mehr selbständig praktisch arbeiten lassen
- Reports aus dem TD aufrufen, Vorstellung des RB

Die vorgesehene Progression des Inhalts orientierte sich an der Praxis: Wesentlich war damals wie heute, dass die Teilnehmer die Prinzipien der objektorientierten Programmierung vollständig kennen, da sie sich ja nach ihrer Rückkehr an ihren Arbeitsplatz in der Regel mit der Weiterentwicklung ihrer Software beschäftigen, die vermutlich auf einem mehr oder weniger komplexen Framework aufsetzt.

Wenn ich mir das sichtbare Ergebnis, also die Bedienoberfläche des Beispiels nach der viertägigen Schulung anschau, dann lässt sich feststellen, dass viele der heutzutage wichtigen visuellen Features für die Anwendungsentwicklung naturgemäß nicht behandelt wurden. Nach vier Tagen hatten die Teilnehmer in vielen Einzelschritten folgendes Ergebnis erreicht, wobei eine Vielzahl von sichtbaren Klassen entstanden sind, die sukzessive verfeinert und verbessert wurden. Das sieht man natürlich nicht.

Besuch in der neuen Videothek

Wie, so habe ich mich gefragt, sähe das Ergebnis einer Neuprogrammierung unter besonderer Berücksichtigung der seitdem neu hinzugekommenen Features und Funktionen aus?

Abbildung 1: das Menü im MDI-Window der Altanwendung

Die Abbildung zeigt (technisch) eine Menüstruktur. Es handelt sich noch nicht um eine Ribbon Bar, die nicht nur statisch, sondern auch programmtechnisch erzeugt und kontextsensitiv eingeblendet und aktiviert werden kann.

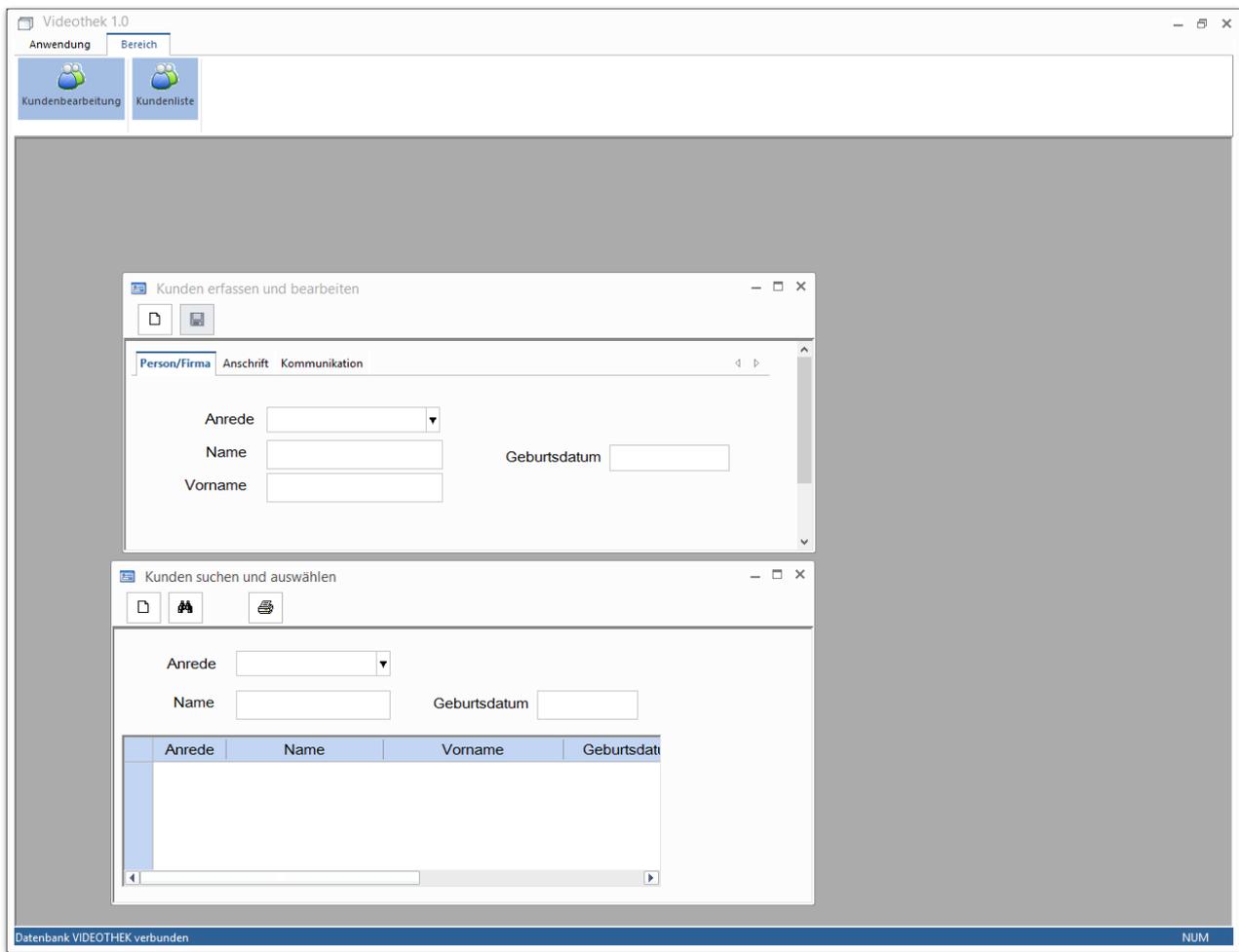


Abbildung 2: das Ergebnis: ein MDI-Window und zwei Bearbeitungsmasken Eingabe und Suchen

Das ist die „klassische“ Fensterstruktur einer SQLWindows-Anwendung: Unter einem MDI-Window können eine Vielzahl von Bearbeitungsfenstern gestartet werden.

Wenn ich mich nur auf die damals nach vier Tagen erreichte Benutzerfunktionalität beschränke, dann sähe das Ergebnis für eine „neue“ Videothekanwendung heute in etwa wie in Abbildung 3 aus:

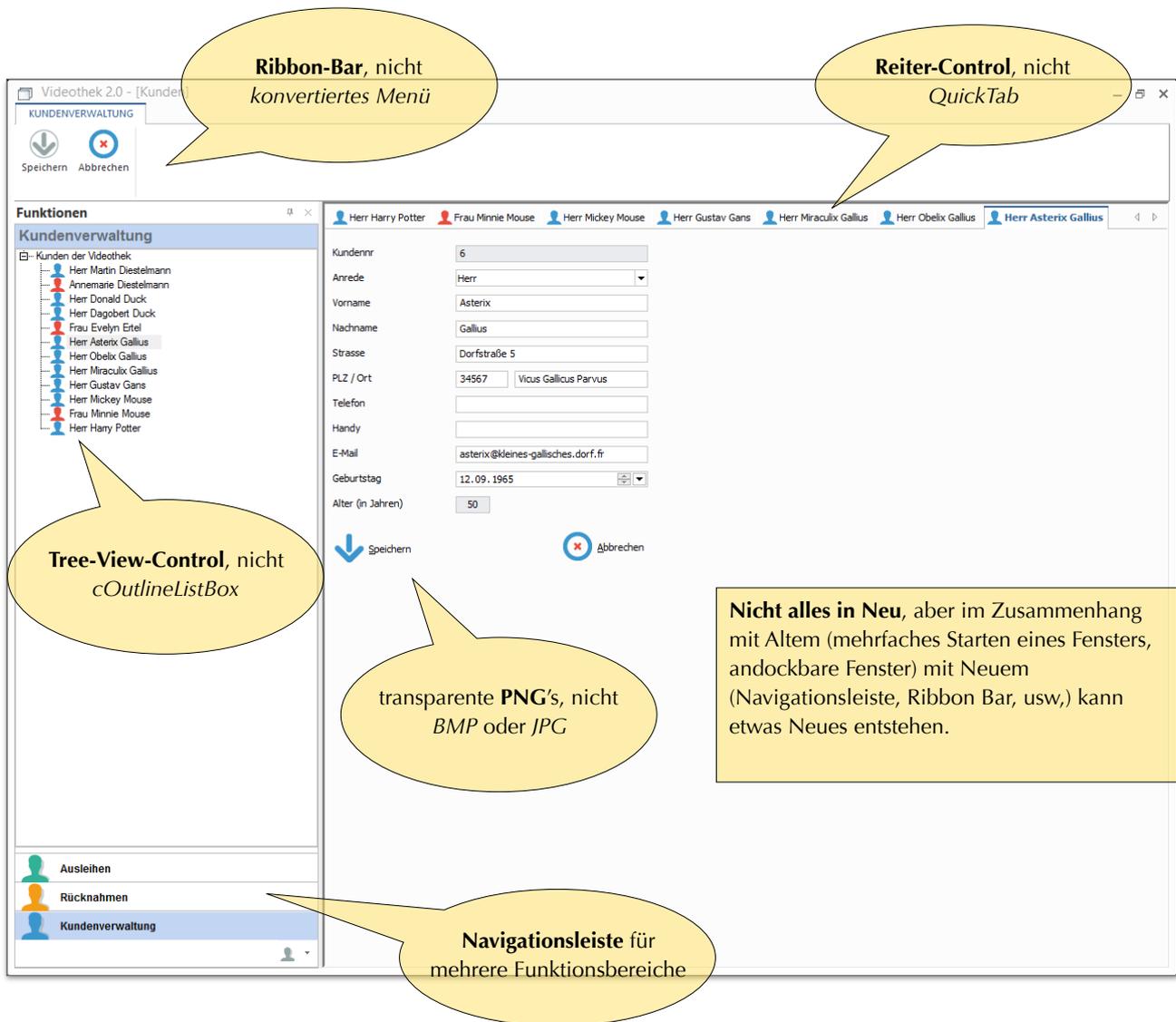
Was ist in der Anwendung gleich geblieben? Nun, die Funktionalität der gesamten Anwendung wird nach wie vor innerhalb eines MDI-Window angeboten. Das ist vermutlich schon alles.

Während in der alten Videothek zwei Fenster im MDI codiert wurden, um die Vorgänge „Erfassen und Ändern eines Kunden“ und „Suche nach einem Kunden“ abzubilden, zeigt die Entwicklung mit den neuen Features ein gänzlich anderes Bild:

- Anstelle des „alten“ MDI-Menüs wurde eine Ribbon-Bar (Menüleiste) implementiert.
- Nach Kunden muss nun nicht mehr gesucht werden, sondern die bestehenden (Beispiel-) Kunden werden sofort angezeigt.

Um das linke Fenster zu realisieren, wurde ein andockbarer Dialog angelegt, aufgerufen und am linken Rand des MDI-Fensters „fixiert“. In dem Dialog befindet sich eine Navigationsleiste, über die ein Wechsel in ein anderes Bearbeitungsfenster (z.B. Ausleihen, Rücknahmen, usw.) angestoßen werden kann. Innerhalb der Navigationsleiste Kundenstamm befindet sich ein Tree-View-Control, in dem bei dieser geringen Datenmenge sofort alle Daten angezeigt werden.

- In der Mitte befindet sich ein Behälter-Fenster (d.h. ein Form-Window mit einer zunächst leeren Reiterleiste). Nach der Auswahl eines Kunden wird das Bearbeitungsfenster des Kunden angezeigt und an die jetzt nicht



Nicht alles in Neu, aber im Zusammenhang mit Altem (mehrfaches Starten eines Fensters, andockbare Fenster) mit Neuem (Navigationsleiste, Ribbon Bar, usw.) kann etwas Neues entstehen.

- mehr leere Reiterleiste gehängt. Es können also mehrere Kundendatensätze gleichzeitig aufgerufen werden, wobei immer nur der Kunde bearbeitet werden kann, in dessen Bearbeitungsfenster sich der aktuelle Fokus befindet. Allerdings kann sehr schnell von einem Kunden zum anderen gewechselt werden, indem der entsprechende Reiter angeklickt wird.
- Insgesamt bieten die neueren Controls viel mehr Möglichkeiten, Images darzustellen, um hiermit die Bedienung für den Anwender einfacher und transparenter zu machen.

Im Ergebnis stelle ich fest, wenn ich das Aussehen und das Verhalten der Beispielanwendungen aus dem Seminar vergleiche, dass in den letzten Jahren eine Vielzahl wesentlicher, sichtbarer Features in der Entwicklungsumgebung hinzugekommen ist.

Es ist aber nicht nur das Aussehen der einzelnen Controls, das sich geändert hat. Zum einen beruht die gesamte Oberflächengestaltung und damit auch das der Controls auf dem Theme einer Anwendung. Zum anderen legt das Zusammenspiel der einzelnen Controls auch eine andere Bedienung nahe, als das möglicherweise in der Vergangenheit implementiert wurde. Während in der Vergangenheit Formulare so gestaltet wurden, dass der Anwender suchen („query by example“ nannte man das) oder eingeben konnte, legen der Einsatz einer Navigationsleiste (möglicherweise im Zusammenhang mit einem Tree-View-Control nahe), bereits in der Startphase eines Fensters dem Arbeitszusammenhang entsprechende Daten anzuzeigen.

Abbildung 3: Entwurf einer „neuen“ Videothek-Software (theming flat design)

Die Anwendung zeigt die Möglichkeiten der aktuellen Team-Developer-Version. Unter einem MDI-Window wird eine Ribbon-Bar angezeigt, über die zunächst der Bearbeitungsbereich gestartet wird. In diesem Fall besteht der Bearbeitungsbereich aus einer Steuerung (links) und einem zunächst leeren Fenster mit einem leeren Reiter-Element. Nach dem Aufruf eines Kunden wird das Bearbeitungsformular für den Kunden gestartet und an das Reiter-Control des Vaterfensters „gehängt“.

Nicht nur das: Durch den Einsatz einer kontextsensitiven Ribbon-Bar sind Aktionen, die früher auf der zweiten Menüebene eingeblendet werden mussten, unmittelbar sicht- und bedienbar.

Fazit

Auch wenn das möglicherweise damals als einleuchtend angesehene Szenario einer Videothek (Jüngere fragen entgeistert: „Was is 'n das?“) möglicherweise heutzutage nicht mehr leicht nachvollzogen werden kann, zeigt die beispielhafte Programmierung im Client-Server-Seminar nicht nur ein moderneres Aussehen.

Es unterscheidet sich hinsichtlich des Bedienkonzepts massiv von einer Implementation, wie man sie möglicherweise vor Jahren vorgenommen hat.

- Die Bedienung moderner Client-Server-Anwendungen ähnelt wesentlich mehr einer datengetriebenen Oberfläche, wie es die Anwender heutzutage auch von populären Internetseiten gewöhnt sind.
- Eine datengetriebene Vorgangsbearbeitung kann mit den heutigen Standardmöglichkeiten von SQLWindows (im Team Developer) wesentlich einfacher und damit schneller realisiert werden.
- Alle hier aufgeführten Bedienelemente gehören zum Lieferumfang von Team Developer. Es werden keine Komponenten anderer Hersteller benötigt.

MD Consulting & Informationsdienste GmbH

Michaelisstraße 13a
99084 Erfurt

Berghamer Straße 14
85435 Erding

phone (+49) 8122 97400
email info@md-consulting.de