

Schnelle Datenbank Anwendungen

In den letzten Ausgaben der Toolbox wurde bereits eine Reihe technischer Beiträge zu Konzept 16 veröffentlicht. Um auch neuen Lesern einen Einstieg in das System zu ermöglichen, sollen die herausragenden Eigenschaften dieses deutschen Entwicklungssystems hier einmal komprimiert dargestellt werden.

von Andrej Mücke

Eine effiziente Entwicklung, eine schnelle Installation und eine durchgängige Administrationsfreiheit kennzeichnen das Datenbanksystem (DBS) Konzept 16 der vectorsoft AG. Das RAD-Tool ist auf die Entwicklung komplexer Softwarelösungen ausgelegt und dabei so einfach in der Handhabung, daß sich nahezu jeder Entwickler schnell damit zurechtfindet.

RAD-Systeme haben den entscheidenden Vorteil, daß sie im Gegensatz zu klassischen Entwicklungsumgebungen aufgrund der Spezialisierung solcher Tools sehr kurze Entwicklungszeiten ermöglichen. Konzept 16 ist jedoch nicht ausschließlich auf kurze Entwicklungszeiten ausgelegt. Mit einer schnellen Installation und einer vollständigen Administrationsfreiheit der auf Basis von Konzept 16 geschriebenen Applikationen sorgt der Software-Hersteller aus Heusenstamm dafür, daß der Aufwand für Vertriebspartner möglichst gering bleibt. Nach diesen Maßgaben sind die Komponenten des Datenbanksystems aufgebaut.

Viel Freiraum für die Gestaltung

Der Designer ist das zentrale Werkzeug in Konzept 16. Er stellt den drei Bereichen Datenbank, Oberfläche und Programmlogik eine Reihe von Editoren und Tools in Fenstern bereit, für die eine globale Drag&Drop-Funktion und kontextsensitive Hilfe vorhanden ist.

Der Benutzer kann alle Fenster je nach Bedarf ein- oder ausblenden und die Umgebung an die Bedürfnisse des aktuellen Entwicklungsstadiums anpassen.

Der Dialogeditor ist für die komplette Benutzeroberfläche zuständig, dafür stehen mehr als fünfzig verschiedene GUI-Objekte zur Verfügung. In einem zusätzlichen Toolfenster sind alle Eigenschaften und Ereignisse des aktiven Objekts sichtbar und editierbar.

Steuerelemente

Der Entwickler kann im ersten Schritt Frames für SDI-, MDI- oder Tray-Anwendungen definieren und diese mit Menüs und Toolbars versehen. Diese Toolbars sind andockbar, verschiebbar und auch als Status- oder Windowbar konfigurierbar. Für Standardfunktionen wie Copy&Paste oder komplexe Objekte wie MDI oder RTF gibt es bereits vordefinierte Toolbars. Mit wenigen Klicks kann beispielsweise ein Dialog um einen RTF-Editor inklusive Symbolleiste erweitert werden, Frames lassen sich dabei ohne weitere Übersetzung sofort testen.

Um den Frame mit Inhalt zu füllen, gibt es Basiselemente wie Standard-Edit-Controls, Labels, Buttons, Groups und Notebooks. Dabei lassen sich Edit-Controls bei Bedarf direkt mit Feldern der Datenbank verknüpfen, was für eine automatische Übertragung von Inhalten zwischen Oberfläche und Daten-

bank sorgt.

Viele Custom-Controls orientieren sich an den Bedürfnissen der GUI-Gestaltung, so zum Beispiel das Tree-View-Objekt, ein Objekt für Gantt-Diagramme, das Picture-Objekt und spezielle Objekte für verschiedene ActiveX-Komponenten, die Office-Anwendungen oder registrierte Datentypen sehr einfach und nahtlos in einen Frame integrieren. Ein besonderes Element ist das *Rec-List*-Objekt; dieses Grid bildet ein Rol-

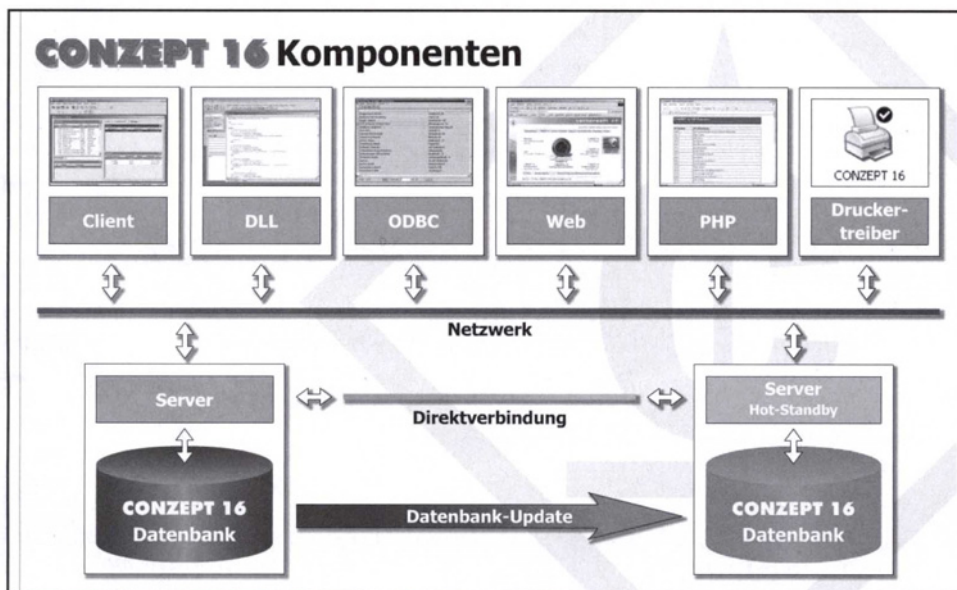


Bild 1: Die Komponenten von Konzept 16

ling Window auf den Datenbestand einer Tabelle ab. Es kann durch ein einziges Drag&Drop vom Datenbankfenster zum Frame definiert werden und ist sofort funktionsfähig. Dabei lassen sich alle Eigenschaften wie beispielsweise Spalten, Sortierung oder Ereignisfunktionen sowohl im Designer als auch zur Laufzeit verändern. Die RecList zeigt die Daten auch bei sehr großen Tabellen sehr schnell an und kann anhand von Datenbankindizes automatisch sortieren.

Der Formulareditor kann unterschiedliche Druckobjekte erstellen und bearbeiten, die die Grundlage für spätere Ausdrucke oder elektronische Dokumente darstellen. Die Basis bildet das PrintDoc-Objekt, das primär das Ausgabeformat festlegt. Neben vierzig vordefinierten Papierformaten sind auch benutzerdefinierte Formate möglich. Um das Objekt mit Daten zu füllen, werden Formularbausteine beliebiger Größe angelegt, die per Drag&Drop mit Texten, Datenfeldern, Bildern, skalierbaren Grafiken, Gantt-Graphen und Barcodes gefüllt werden.

Diese Elemente lassen sich innerhalb des Objekts frei positionieren und enthalten zahlreiche Eigenschaften wie Schriftart, Farbe oder Ausrichtung, die sich auch während der Laufzeit des Druckvorgangs noch verändern lassen. Ein anpaßbares Druckvorschau-Objekt für die Oberfläche kann selbst Druckjobs mit mehreren tausend Seiten Umfang anzeigen.

Über das Datenbank-Info-Tool hat der Entwickler jederzeit Zugriff auf das komplette Data Dictionary. Der Daten-

bank-Editor gestattet darüber hinaus direkte Änderungen der Datenbanktabellen inklusive des Indexaufbaus.

An C angelehnte Programmiersprache

Die Programmlogik der Anwendung wird in Form von Prozeduren in einer Concept-16-eigenen Sprache geschrieben, die sich an C anlehnt, der Sprachumfang orientiert sich an den Anwendungsschwerpunkten Datenbankverarbeitung und Oberflächensteuerung. Wesentliche Elemente wie Subfunktionen, dynamische Variable und Kontrollstrukturen sind natürlich enthalten. Für die Fehlerbehandlung gibt es Exceptions in Form von try-Blöcken.

Der Prozedurtext wird in einen portablen binären P-Code übersetzt, der auf unterschiedlichen Plattformen ausgeführt werden kann.

Durch dynamisches Binden kann der Code ohne zusätzlichen Linklauf sofort ausgeführt werden. Potentielle Laufzeitfehler durch Late Binding kann der Entwickler mit einer speziellen Testfunktion vermeiden.

Der Prozedureditor im Designer verfügt über Syntax-Highlighting, Helptips und direkte Befehlshilfen. Breakpoints und Hotspots werden direkt im Quelltext gesetzt. Bei einem Übersetzungsfehler wird automatisch der Fehlertext angezeigt und auf die entsprechende Stelle in der Prozedur positioniert. Zusätzlichen Komfort bietet eine globale Suchfunktion über alle Prozeduren der Datenbank. Die Befehlshilfe kann um eigene Programmdokumentationen er-

Datenbanken

weitert werden, weshalb Beschreibungstexte für eigene Funktionen oder Variablen integriert werden können.

Eine Reihe fest eingebauter Bibliotheken stellt eine große Zahl von Befehlen für unterschiedliche Programmbereiche zur Verfügung, beispielsweise für mathematische Berechnungen, String-, Datums- und Zeitverarbeitung sowie dynamische Strukturen wie beispielsweise verkettete oder sortierte Listen.

Programmbibliotheken

Im Bereich der Datenbankverarbeitung hat eine Prozedur zunächst Zugriff auf das Data Dictionary, das komplette Benutzersystem und auf administrative Datenbankfunktionen. Den Schwerpunkt bilden jedoch die Befehle für den Zugriff auf Tabellen, die Transaktionsverarbeitung und die Generierung und Durchführung von Abfragen auf den Datenbestand.

Die Funktionen für das GUI erlauben ein Browsing der Objektstruktur der Frames und ihrer Elemente. Damit kann die Prozedur zur Laufzeit für eine dynamische Verarbeitung alle relevanten Daten der Oberfläche ermitteln, außerdem kann sie alle Eigenschaften der GUI-Objekte zur Laufzeit verändern. Aufgerufen werden die Steuerungsprozeduren durch Ereignisse, wobei die Zuordnung von Prozeduren und Ereignissen modifizierbar ist. Im Extremfall läßt sich damit ein leistungsfähiges, dynamisches Event-Routing verwirklichen.

Weitere Bibliotheken stehen für unterschiedliche Schnittstellen bereit. So lassen sich COM-basierte Anwendungen genauso einfach ansprechen wie aus anderen Systemen. Darüber hinaus wird die Kommunikation über Sockets und SMTP, die Verbindung zu Telefonsystemen mit TAPI und der Zugriff auf andere Datenbanken über ODBC ermöglicht. Zusätzliche Erweiterungen sind durch eigene DLLs möglich.

Alle im Designer angelegten Tabellen, Objekte und Prozeduren werden in einer Entwicklungsdatenbank gespeichert, die wiederum aus einer einzigen Datei besteht. Die Datenbankdatei ist für alle Plattformen binärkompatibel. Die gesamte Anwendung kann daher sehr einfach gesichert oder transportiert werden.

Der Entwickler kann Anwendungselemente jederzeit einzeln oder als Gruppe exportieren oder importieren. Die Entwicklungsdatenbank dient später auch als Grundlage für die Installation beim Anwender.

Windows oder Linux

Den Datenbankserver von Konzept 16 es als 32- und 64-Bit-Version für Windows- und Linux-Systeme. Jede Datenbank wird als eigener Prozeß ausgeführt, in dem für jeden User exklusiv ein eigener Thread bereitsteht. Die 64-Bit-Variante kann dabei pro Datenbank bis zu 16 GByte des verfügbaren Hauptspeichers als Cache nutzen. Der Server kennt geschlossene geschachtelte Transaktionen, Transaktions-Logging und verfügt über eingebaute Recover-, Diagnose- und Repair-Funktionen. Konfiguriert wird der Server über einen Webbrowser, wobei weder ein zusätzlicher Webserver noch aktive Komponenten wie ActiveX oder Java benötigt werden.

Backup-Events bringen eine Datenbank vorübergehend in einen konsistenten Read-only-Zustand; da die angemeldeten Benutzer normal weiterarbeiten können, ist ein 24-

Stunden-Betrieb realisierbar. Backup-Events können geplant oder dynamisch per Kommandozeilen-Utility ausgelöst werden.

Der Betrieb des Datenbankservers ist auf Wartungsfreiheit ausgelegt. Der Server kann täglich die Datenbank automatisch untersuchen, um Problemsituationen frühzeitig zu erkennen. Eine E-Mail-Benachrichtigung sorgt im Fehlerfall für eine umgehende Alarmierung der Systemverantwortlichen. Weitere Hilfsprogramme erlauben das direkte Abholen der Protokolldateien von entfernten Servern und die gezielte Auswertung der Datenbank-Protokolle.

Betrieben wird die Applikation mit einem Clientprogramm, das den Datenbank-Connector, das P-Code-System und die Bibliotheken in sich vereint. Der Client läuft auf den Betriebssystemen von Windows 98 bis Vista, wobei plattformsspezifische Eigenschaften wie die Styles oder Themes automatisch angepaßt werden.

Neben dem Standard-GUI-Client gibt es weitere Clients, die im Kern auch jeweils den Datenbank-Connector und das P-Code-System enthalten. So kann der Service-Client beispielsweise Prozeduren im Kontext eines Windows-Dienstes durchführen, der DLL-Client erlaubt den Zugriff auf die Datenbank und das Ausführen von Prozeduren von allen Programmen aus, die DLLs einbinden können. Daneben gibt es noch spezielle Varianten für PHP und ein Webserver-Plugin sowie einen universellen ODBC-Treiber für die Datenbank.

Zu den Entwicklungswerkzeugen gehört auch ein Source-Level-Debugger mit Variablenüberwachung, Breakpoints, Single-Step-Betrieb und Zugriff auf das Data Dictionary der Datenbank. Grundsätzlich gibt es keinen Unterschied zwischen Debug- und Release-Code der Prozeduren, da bei der Übersetzung einer Prozedur alle zum Debuggen benötigten Informationen separat in der Datenbank abgelegt werden.

Kommuniziert wird zwischen P-Code-System und Debugger über TCP/IP, weshalb der Debugger auf einem Remote-System ausgeführt werden kann.

Installiert werden Datenbankserver und Clients mit einem einzigen Setup-Programm, in dem bereits alle Komponenten enthalten sind. Die meisten Clients können ausschließlich auf dem Server eingerichtet werden, eine einfache Verknüpfung genügt dann zum Start der Applikation auf den Arbeitsplatzrechnern.

Einfache Installation der Applikationen

Aktualisiert werden die installierten Komponenten in der gleichen Art und Weise: Es wird eine spezielle Definitionsdatei mit Anwendungselementen der Entwicklungsdatenbank angelegt, sie wird dann in die produktive Datenbank eingespielt. Zusätzliche Aufgaben wie beispielsweise den Aufbau neuer Indizes kann eine anschließende spezielle Prozedur erledigen.

Für den Konzept-16-Datenbankserver gibt es eine zusätzliche hardwareunabhängige Hochverfügbarkeitslösung. Diese Hot-Standby-Option benötigt einen zweiten Serverrechner mit eigenem Stagesystem zur kontinuierlichen Datenreplikation über eine separate Standard-Netzwerkverbindung.

Die beiden Datenbankserver müssen nicht identisch sein, auch ein gemischter Betrieb mit unterschiedlichen Betriebssystemen auf beiden Servern ist möglich. ■