

Die Aufgabe bestand darin, das Programm LEMON von Oracle Forms 6i nach .NET zu konvertieren. Dies war notwendig, da das verwendete Oracle Graphics nicht mehr weiterentwickelt wird und das Programm deshalb bald nicht mehr angeboten werden kann. Das Ziel des Projekts war es, einen Prototyp zu erstellen, der die wichtigsten Komponenten dieses Programms umsetzt und dadurch dem Unternehmen den zukünftigen Weg weist. Die umzusetzenden Komponenten sollen möglichst identisch – wie in der bisherigen Form – aussehen, damit beim Kunden ein Umstieg kaum spürbar ist.



LEMON goes C#

Armin Pristauz, Alexander Armbruckner, Florian Kalliauer und Dipl.-Ing. Peter Anzenberger, HTL Grieskirchen

Die vorliegende Arbeit wurde im Rahmen einer Diplomarbeit an einer österreichischen Höheren Technischen Lehranstalt (HTL) für EDV und Organisation durchgeführt. Es handelt sich damit um die Abschlussarbeit zum Abitur. Diese Arbeiten werden in den meisten Fällen für Firmen durchgeführt und von den Schülern weitgehend selbstständig abgewickelt. Von der Schule werden Fachlehrer nominiert, die sich um die Betreuung und Bewertung solcher Arbeiten kümmern. Das Besondere daran ist, dass es sich um Aufgabenstellungen handelt, die oft in dieser Form im Unterricht nicht explizit behandelt werden. Oracle Forms, Graphics und Reports werden nur sehr marginal im Unterricht verwendet, deshalb war die erste Herausforderung für die Schüler, diese Werkzeuge kennenzulernen und so weit zu verstehen, dass eine Umsetzung mit neuen Technologien möglich ist.

Das Projekt wurde im Juni 2009 mit einer grundsätzlichen Ideenfindungsphase gestartet. Während des Sommers standen das Kennenlernen der Oracle-Tools und das Schreiben des Pflichtenheftes auf dem Plan. Dieses Pflichtenheft wurde im Rahmen einer Feriapraxis eines Teammitglieds bei der Firma UTAS erstellt.

Die Programmierung begann im September 2009. Wir entschieden uns für einen agilen Ansatz und damit gegen das klassische Software-Engineering. Zur Diskussion stand, ob wir die agile Methode SCRUM anwenden oder generell auf einen bekannten Ansatz verzichten sollten, weil wir sowieso schon regelmäßige Besprechungen mit unserem Betreuungslehrer geplant

hatten. Wir entschieden uns letztendlich dafür, auf eine bekannte agile Methode zu verzichten, auch, weil so eine Methode ohnehin schwer mit einem Drei-Mann-Team umzusetzen gewesen wäre. Außerdem sollten die drei Module vollkommen selbstständig arbeiten und auch an jeder anderen Stelle eingesetzt werden können. Wir mussten uns also hinsichtlich dessen lediglich in Sachen Schnittstellen absprechen.

Das Ziel bei der grafischen Gestaltung des Emissionsstatus' in „LEMON goes C#“ war es, den bisher in Forms umgesetzten Emissionsstatus mit so wenig grafischen Änderungen wie möglich neu zu implementieren. Früher wurde ein Emissionsstatus für jeden Kunden manuell im Forms-Editor angefertigt. Das hatte den Vorteil, dass sich Ansichten speziell auf Kundenwünsche anpassen ließen, aber auch den Nachteil, dass solch eine Anpassung mit einem großen Aufwand verbunden war.

Im Gegensatz zu Forms erfolgt die grafische Gestaltung in GDI+ allerdings nicht mit einem eigenen Editor, sondern ist nur direkt im C#-Code möglich. Das erlaubt uns eine viel dynamischere Gestaltung als im originalen LEMON-System, in dem zum Beispiel auch nur vier verschiedene Anlagen zur selben Zeit angezeigt werden konnten. Daraus entwickelte sich die dynamische Anzeige des Emissionsstatus', bei dem die Ansicht je nach Anzahl der Anlagen und der damit verbundenen Komponenten automatisch skaliert wird und der sich sehr stark am ursprünglichen Bild des Emissionsstatus' von LEMON orientiert. Trotzdem fehlen dieser Ansicht einige Feinheiten

und individuelle Anpassungen sind nur sehr beschränkt möglich.

Als Lösung wurde schließlich der LEMON-Emissionsstatus-Editor entwickelt. Dieses unabhängige Programm erlaubt die individuelle Gestaltung eines Templates für einen Emissionsstatus, definiert die Positionen, Größen

Die Firma UTAS (Umwelt-Technik-Anlagenplanung-Software) ist in Linz ansässig, seit über 15 Jahren am Markt und eines der marktführenden Unternehmen im deutschsprachigen Raum für Software zur kontinuierlichen Umweltüberwachung, vor allem im Bereich der Luft- und Abwasseremissionen. Zu den Kunden von UTAS zählen über 100 Unternehmungen der Mittel- und Großindustrie, wie z.B. die voestalpine STAHL, Austria Metall AG sowie die meisten Zementwerke und Müllverbrennungsanlagen in Österreich.

Das Abgasüberwachungssystem LEMON steht für „Luft-Emissions-Monitoring“ und besteht aus zwei Teilen – LEMESS (Luft-Emissions-Messrechner) und LEMON. LEMESS steht in direkter Verbindung mit verschiedenen Sensoren, misst sämtliche Emissionen und schreibt die erfassten Werte in die Datenbank. LEMON kümmert sich um die Visualisierung und Auswertung. Dazu liest es die Daten aus der Datenbank, generiert verschiedene grafische Statistiken und erzeugt Berichte. Das LEMON-System ist vom Benutzer jederzeit und vollständig parametrierbar.

etc. der grafischen Elemente und wird später mit aktuellen Daten gefüllt.

Ein großes Problem war, die umfangreiche Datenmenge dynamisch und schnell darzustellen. Zuerst probierten wir es mit einer normalen HTML-Tabelle. Dabei hatten wir jedoch Probleme, die Datenmengen dynamisch zu beschränken (zum Beispiel möchte ein Benutzer zehn Reihen sehen, der andere wieder zwanzig). Das war natürlich mit HTML nur schwer machbar. Deshalb haben wir uns für das jQgrid entschieden. Damit ist die Darstellung ganz einfach; der Benutzer kann selber parametrieren, welche Daten, wie viele und wie er sie angezeigt haben will.

Bei der Projektplanung gab es zunächst Windows Forms, Windows Presentation Foundation (WPF) sowie die grafische Darstellung in Java als mögliche Alternativen. Obwohl die Entwicklung der grafischen Darstellung in Java besonders in Lizenzfragen unser Projekt vereinfacht hätte, fiel die Entscheidung relativ früh gegen Java, da die Firma UTAS schon Teile ihres Produktsortiments auf .Net-Basis neu implementiert hatte und somit eine weitere Technologie integrieren müsste.

Als realistische Alternativen blieben somit nur noch Windows Forms und WPF. Für WPF sprach zu diesem Zeitpunkt vor allem die Kompatibilität zwischen Web- und Client-Anwendungen, da sich WPF-Projekte mühelos von einer Client- zu einer Web-Anwendung konvertieren lassen. Für diese WPF-Web-Anwendungen kommt dabei die Technologie „Silverlight“ zum Einsatz, die eine reduzierte Version der WPF-Bibliothek darstellt. Silverlight besitzt jedoch den entscheidenden Nachteil, dass – ähnlich wie bei Flash – der ausführende Client erst die Technologie für seinen Browser installieren muss.

Bei der Auswahl einer geeigneten Technologie für den Datenbank-Zugriff in dem Projekt galt es, verschiedene Anforderungen zu erfüllen:

- Gute Unterstützung von Oracle-Datenbanken
- Gute Performance beim Abfragen und Sortieren vieler Datensätze
- Die Abfragen müssen auf SQL basieren

Als Alternativen boten sich zuerst ADO.NET, Language Integrated Query (LINQ) to SQL sowie spezifische ADO.NET-Data-Provider für Oracle an. LINQ to SQL stellt dabei zwar zweifelsohne die jüngste Technologie dar, besitzt allerdings den Nachteil, dass sie eher für kleinere Datenbanken und Abfragen entworfen wurde und keine gewöhnliche SQL-Syntax für Statements benutzt. Somit blieb nur noch ADO.NET als geeignete Technologie übrig, welche sich zudem als offizielle Entwicklung von Oracle anbot. Auch im Vergleich mit anderen Data-Providern für Oracle schnitt ODP.NET aus Sicht der Autoren am besten ab.

.NET 3.5 & C#

Für das Projekt kam die aktuelle Version des .NET-Frameworks in der Version 3.5 zum Einsatz. Zur Diskussion standen C# und Java wegen der Plattformunabhängigkeit. Da UTAS diese nicht unbedingt benötigt und LEMON bisher immer eine Windows-Anwendung war, einigte man sich auf C#.

.NET-Framework 3.5

Das .NET-Framework ist eine Plattform zur Software-Entwicklung mit verschiedenen Klassenbibliotheken sowie Programmierschnittstellen (API). Der Funktionsumfang von .NET umfasst neben der Web-Entwicklung auch die Gestaltung grafischer Client-Anwendungen und die Entwicklung für mobile Endgeräte. Ein wesentliches Merkmal des .Net-Frameworks ist die Implementierung des Common Language Infrastructure-Standards (CLI), der eine gemeinsame Basis für die Entwicklung von Programmen mit verschiedenen Sprachen und Plattformen bietet. Diese Universalität wird auch in der Laufzeitumgebung durch die Common Language Runtime (CLR) gewahrt, indem die Common Intermediate Language (CIL) vom Interpreter benutzt wird, welche es ermöglicht, dass verschiedene Hochsprachen den gleichen CIL-Bytecode erstellen.

Mittlerweile existiert auch eine relative Plattformunabhängigkeit innerhalb des .NET-Frameworks, etwa durch das

Mono-Projekt, das eine Unterstützung des .NET-Frameworks auf Linux bietet. Allerdings sind die meisten dieser Konvertierungen nicht auf dem aktuellen Stand von .NET und somit nur eine beschränkte Alternative zur Entwicklung auf Basis von Microsoft Windows.

C#

C# ist eine von Microsoft entwickelte, objektorientierte Programmiersprache und Bestandteil des .NET-Frameworks. Die Architektur und das Design von C# besitzen große Ähnlichkeit mit der ebenfalls objektorientierten Sprache Java; Microsoft selbst sieht C# allerdings eher an C++ angelehnt. Als Entwicklungsumgebungen für C# existieren die frei verfügbaren Programme Sharp-Develop, MonoDevelop sowie Microsoft Visual Studio Express Edition.

ADO.NET

ADO.NET ist eine Sammlung von Klassen, welche den Zugriff auf relationale Datenbanken ermöglicht. ADO.NET basiert auf der Technologie ActiveX Data Objects (ADO), bietet jedoch wesentliche Verbesserungen zu ADO 2.x, sodass ADO.NET im Prinzip eine Neuentwicklung darstellt. Wesentliche Merkmale sind die universelle Schnittstelle für alle .NET-Sprachen sowie die drei Hauptklassen „DataSet“, „DataTable“ und „DataReader“, die das RecordSet unter ADO ablösen.

Die Schnittstelle zur Datenbank stellt der Data-Provider dar. Dabei handelt es sich um eine Art Treiber, der individuell an die jeweilige Datenbank angepasst ist (der Oracle-Data-Provider wird später noch genauer behandelt). Um die DataSets zu befüllen, Daten aus der Datenbank zu lesen, zu verändern und zu speichern, bedient sich der Data-Provider wiederum des Data-Adapters, der die Kommunikation mit der Datenbank behandelt.

ODP.NET

Der Oracle-Data-Provider for .NET (ODP.NET) stellt einen ADO.NET-Data-Provider für Oracle-Datenbanken dar und ist von Oracle selbst entwickelt worden.

Zusätzlich zu allen Funktionen, die ADO.NET standardmäßig bietet, unterstützt ODP.NET auch noch Oracle spezifische Features und besitzt eine enge Bindung an die Oracle Datenbank, was beispielsweise die Sicherheit erhöht.

Wichtige Vorteile von ODP.NET gegenüber anderen Data-Providern sind unter anderem auch Connection Pooling, also die Wiederverwendung physischer Datenbankverbindungen, Statement Caching, welches die Performance durch das Cachen oft verwendeter Statements (z. B. bei Loops) erhöht, sowie 64Bit-Support.

ASP.NET

Da unser Auftraggeber plant, LEMON zu einer serverseitigen Web-Applikation zu machen, verwenden wir ASP.NET (Active Server Pages .NET) als Testumgebung. Das heißt, wir müssen keine fertige Website erstellen, sondern es soll nur ein mögliches neues Aussehen von LEMON aufgezeigt werden.

ASP.NET basiert auf dem .NET-Framework und dadurch ist es theoretisch möglich, Webanwendungen in allen CLR-kompatiblen Sprachen zu erstellen, also auch in C#. Die Version 3.5 von ASP.NET wurde um AJAX (ermöglicht asynchrone Datenübertragung zwischen Server und Browser) und LINQ (dient zur Abfrage von Datenquellen wie Datenbanken) erweitert.

ASP.NET MVC

ASP.NET MVC ist ein Framework von Microsoft. MVC steht für „Model-View-Controller“. Das ist ein Software-Pattern, das die Logik von der Präsentation trennt. Models sind in Klassen gefasste Daten. Die Seite wandelt das Model in eine geeignete Form, um diese optimal anzuzeigen. Der Controller empfängt die Daten, berechnet daraus die Aktion und antwortet dann mittels der Model-Objekte. Das ASP.NET MVC-Framework verbindet diese drei Komponenten miteinander. Sie können aber jeweils einzeln getestet werden, was die Fehlersuche erleichtert. Außerdem ist das URL-Mapping unter MVC sehr leistungsstark, da man da-

```
string oradb = „Data Source=(DESCRIPTION=(ADDRESS_LIST=“
+ „(ADDRESS=(PROTOCOL=TCP)(HOST=ORASVR)(PORT=1521)))“
+ „(CONNECT_DATA=(SERVER=DEDICATED) (SERVICE_NAME=ORCL));“
+ „User Id=scott;Password=tiger;“;
Connection conn = new OracleConnection(oradb);
conn.Open ();
OracleCommand ocmd = new OracleCommand();
ocmd.Connection = conn;
ocmd.CommandText = „select deptno, dname, loc from dept“;
ocmd.CommandType = CommandType.Text;
OracleDataReader odr = ocmd.ExecuteReader(); while (odr.Read())
{
    listBox1.Items.Add („The „ + odr[„dname“].ToString() +
        „ department is in „ + odr[„loc“].ToString());
}
```

Abbildung 1: ODP.NET-Zugriff in C#

mit sehr einfach zwischen verschiedenen Actions wechseln kann.

AJAX

AJAX ist ein System der asynchronen Datenübertragung zwischen Browser und Server. Es können, während die Hypertext Markup Language (HTML)-Seite angezeigt wird, nebenbei noch Hypertext Transfer Protocol (HTTP)-Anfragen durchgeführt werden, wobei die Seite nicht neu geladen werden muss. Außerdem kann man die Seite auch komplett verändern, ohne diese neu laden zu müssen.

jQuery

jQuery ist ein frei verfügbares JavaScript-Framework, welches über sehr umfangreiche und mächtige Funktionen zur Navigation und Manipulation der Document Object Model (DOM)-Syntax des DOM bereitstellt. DOM ist eine Spezifikation einer Schnittstelle für den Zugriff auf HTML- oder XML-Dokumente. Die vereinfachte Navigation und die einfache Einbindung in die Website gehören zu den Stärken von jQuery. Des Weiteren bietet das Framework elegante und leicht verständliche Funktionen für animierte Effekte, AJAX und Event-Handling. Auf unserer Website verwenden wir die Version 1.2.2.

GDI+

Zum Zeichnen der Grafiken, welche wir in ASP.NET anzeigen, verwenden

wir Graphics Device Interface (GDI+). Dies ist der Nachfolger von GDI und bietet neue Features zur nativen Darstellung von JPEG und PNG. Besonders für die Darstellung von Grafiken auf der Webseite war dieses Feature in unserem Diplomprojekt von Bedeutung. Neben C# wird GDI+ außerdem in C sowie C++ unterstützt, beziehungsweise in allen Windows-Anwendungen. Für die Verwendung von GDI+ müssen auch keine zusätzlichen Komponenten installiert werden, da GDI+ standardmäßig ab Windows XP in allen Windows-Betriebssystemen enthalten ist.

iTextSharp

iTextSharp ist eine C#-Klassenbibliothek und wurde zur Erstellung der Berichte eingesetzt. Es wurde ursprünglich für Java entwickelt und dient der Erzeugung und Bearbeitung von PDF-, HTML- und „Rich Text Format“-Dateien. Mit der Portierung iTextSharp ist es uns möglich, den vollen Funktionsumfang auch in C# zu nutzen. Lizenziert ist diese Bibliothek unter der GNU „Affero General Public License“ (AGPL). iTextSharp ist besonders hilfreich, wenn man sehr große und Benutzereingabe-abhängige PDF-Dokumente erzeugen muss, so wie in unserer Diplomarbeit. Ein Tagesbericht umfasst nicht selten über 25 Seiten mit Tausenden von Werten.

Fazit

Ein Projekt dieser Art ist sicherlich eine große Herausforderung für Absolven-

ten einer HTL. Das Kennenlernen einer relativ unbekanntem Technologie, wie es die Oracle Devoper Suite darstellt, und die Umsetzung in eine neue Umgebung sind gewiss keine leichten Aufgaben. Vor allem, wenn es sich um ein bereits eingeführtes Produkt handelt und damit auch das Layout nicht frei gewählt werden kann. Für viele Firmen stellt allerdings die Umstellung von Forms auf die Application-Server-Run-time-Umgebung keine realisierbare Alternative dar. Das vorliegende Projekt sollte für die Firma UTAS zeigen, dass eine Alternative zu Forms 6i möglich ist. Und schlussendlich war die Firma UTAS froh, das Projekt so erfolgreich realisiert zu haben. Diese neue Version wird bereits bei zahlreichen in- und ausländischen Kunden eingesetzt und jetzt von Mitarbeitern der Firma UTAS weiterentwickelt.

Kontakt

Dipl.-Ing. Peter Anzenberger
 p.anzenberger@htl-grieskirchen.at



Christian Schwitalla
 Leiter SIG Development



Mehr zum Thema „Forms“

Die riesige Resonanz auf unseren Aufruf und die ungebrochene Beliebtheit von Forms haben den Rahmen dieser DOAG News gesprengt. Es gingen mehr Beiträge ein, als wir Platz dafür hatten. Aus diesem Grund sind in der nächsten Ausgabe weitere Forms-Artikel abgedruckt.

Auch die Veranstaltung der Special Interest Group Development am 5. April 2011 in Frankfurt beschäftigt sich komplett mit der Modernisierung und Migration von Forms-Applikationen. Alle Teilnehmer sowie alle DOAG-Mitglieder können sich die Vorträge herunterladen.

Zusätzlich findet am 10. bis 11. Mai 2011 das Berliner Expertenseminar von Gerd Volberg zum Thema „How to create a Forms-Framework“ statt.

Weitere Informationen unter www.doag.org/go/forms



MuniQSoft GmbH – Datenbanken mit iQ

IT-Consulting	Schulungen	Software-Lösungen	Oracle Lizenzen
<ul style="list-style-type: none"> › Performance Tuning <ul style="list-style-type: none"> • Oracle Datenbank Tuning • Oracle SQL + PL/SQL Tuning › Real Application Clusters › Data Guard + Fail Safe › Datenbank Management <ul style="list-style-type: none"> • Konfiguration • Backup & Recovery • Migration und Upgrade › OEM Grid Control › Oracle Security › Services <ul style="list-style-type: none"> • Remote DBA Services • Telefon-/Remotesupport 	<ul style="list-style-type: none"> › Oracle SQL › Oracle PL/SQL › Oracle DBA › Oracle APEX › Backup & Recovery › RMAN › Neuerungen 10g/11g › Datenbank Tuning › Datenbank Monitoring › Datenbank Security Wir bieten Ihnen öffentliche Kurse sowie Inhouse-Schulungen. 	<ul style="list-style-type: none"> › Individualsoftware <ul style="list-style-type: none"> • .NET und Visual Basic • Java › Oracle APEX › PL/SQL Unser Ziel: Individuelle Softwareentwicklung mit Fokus auf Ihre Zufriedenheit. 	<ul style="list-style-type: none"> › Oracle Datenbanken <ul style="list-style-type: none"> • Standard Edition One • Standard Edition • Enterprise Edition • Personal Edition › Oracle Produkte <ul style="list-style-type: none"> • Enterprise Manager • Oracle Tools Optimale Lizenzierung durch individuelle Beratung.

Nutzen Sie unsere Kompetenz für Ihre Oracle Datenbanken.

