



Midrange MAGAZIN

Ausgabe Mai 2005

RPG unter .Net – ASNA VisualRPG 7.1

Zukunft des RPG?

IBMs Empfehlung, Projekte unter Java und Websphere zu entwickeln, war für viele keine Option. Doch die Zeit drängt. Vielleicht ist VisualRPG.Net von ASNA die Lösung?

Die IBM-Rechnerfamilie S/3x existiert seit den 70er Jahren. Seitdem hat sie sich unter verschiedenen Namen präsentiert: AS/400, später iSeries und nun i5. Der Name änderte sich, doch die Anwendungsprogramme dieser Maschinen zeigten sich von der 30-jährigen Entwicklung der IT weitgehend unbeeindruckt. Die Anwender waren kaum zur Erneuerung gezwungen. Vielfach blieben auch die Software-Entwickler auf dem einmal erarbeiteten Wissensstand stehen. Durch den aufgestauten Erneuerungsdruck stehen viele Unternehmen vor der Frage, wie sich ihre IT weiterentwickeln kann. Die meisten betreiben neben i5 auch Windows-Server und haben festgestellt, dass Windows-Systeme mittlerweile ebenso stabil laufen wie die i5. Bei vielen Anwendern sind Windows- oder Web-Projekte im Einsatz, die i5-Daten verwenden. Doch die Integration von Native-i5-Daten auf Windows ist nicht so einfach.

RPG und Java

Ein Großteil der Projekte auf der i5 ist in der Sprache RPG erstellt. IBM legt den Anwendern seit Jahren einen Umstieg auf Java/Websphere nahe. Für viele Unternehmen ist eine Migration jedoch zu teuer, zumal die Folgekosten oft nicht abzusehen sind. Einen wesentlichen Anteil daran haben die Schulungskosten für die IT. Auch der lange Produktionsausfall bei Umstieg auf Java ist zu berücksichtigen, denn die Entwicklung von Projekten in Java unterscheidet sich erheblich von der RPG-Programmierung.

In dieser Situation bietet die .Net-Plattform zusammen mit den Produkten des IBM- und Microsoft-Partners ASNA eine interessante Alternative. Dieses texanische Software-Unternehmen entwickelt seit den 80er Jahren Software für die Midrange-Community, auch für IBM und konnte viele große Midrange-Anwender für ihre Plattform gewinnen. Als strategischer Partner von Microsoft entwickelte das Unternehmen die Sprache Visual RPG .Net. Derzeit ist Version 7.1 verfügbar. Die Bibliotheken aus Visual RPG stehen auch C#- und VB.Net-Programmierern zur Verfügung. Für die kommenden Versionen von Visual Studio (Whidbey) und SQL Server (Yukon) wurde Version 8.0 angekündigt.

ASNA VisualRPG.Net

Der RPG-Compiler integriert sich in Visual Studio .Net 2003 und führt RPG, Web und .Net unter Windows zusammen. Die Verbindung zu Datenbanken erfolgt durch ASNAs DataGate.

DataGate und die Datenbanken DB2/400, MS-SQL-Server und Acceler8DB von ASNA

Die Middleware bietet einen direkten Lese-/Schreibzugriff auf Satzebene und abstrahiert somit den Datenbankzugriff von der Datenbank selbst. SQL oder ODBC ist dabei nicht notwendig, kann aber eingesetzt werden. Das Übertragungsprotokoll ist TCP/IP.

.Net ist nicht der Feind von RPG

Die Sprache RPG hat im Laufe ihrer Geschichte Merkmale angesammelt, die nicht in eine zeitgemäße Umgebung passen.

Deshalb kann man kein RPG-Programm von i5 in VisualRPG reinkopieren und erwarten dass es läuft.

Die Syntax von VisualRPG orientiert sich an FREE-RPG und CL. Es wurden nicht nur die bekannten OpCodes erstellt, neue OpCodes und BuiltInFunctions erleichtern dem RPG-Programmierer das Leben auf .Net.

Wesentlich ist, dass es auf der Datenbank keinen Unterschied macht, ob der CHAIN, WRITE oder UPDATE von i5/RPG oder VisualRPG aufgerufen wird. Wirft ein RPG-Programmierer einen ersten Blick auf den Code von VisualRPG ist die übliche Reaktion die Frage, wo denn hier RPG ist. Die Antwort ist ‚Inside‘. Der ‚Trick‘ von VisualRPG ist, dass der Compiler i5/API-Calls produziert, die an DataGate übergeben werden. Der DataGate-Job läuft als Batch-Job unter dem User, der am Client arbeitet. DataGate übergibt den vom Client empfangenen Aufruf an i5/OS und sendet das ResultSet an den Aufrufer zurück. Dadurch ist die Verarbeitungsgeschwindigkeit gleich der eines Native-Dialogprogramms.

Die Syntax von VisualRPG

F-Specs heißen DclDiskFile:

DclDiskFile

```
Name(CMMastF) +
DB(LocalDB) +
File + ("LIBL/CMAstNewL2") +
Mbr(*First) +
Org(*Indexed) +
Type(*Update) +
ImpOpen(*No)
```

Name: Name unter dem der File im Programm angesprochen wird.
DB: Der Verweis auf das Datenbank-Objekt. Im DB-Objekt wird die Datenbank mit IP-Adresse, Bibliotheksliste etc. spezifiziert.
File: Der tatsächliche Name des Files. Im Beispiel ist ein Textstring angegeben, der natürlich mit einer Variablen überschrieben werden kann.
Mbr: Teildateiname
Org: Dateorganisation
Type: öffnen für Input
ImpOpen: Implizites Open mit *NO verhindert; hier wird die Datei im Programm mit OPEN geöffnet.

Chain in VisualRPG:

```
Chain CMMastF +
Key( inCustNo ) +
Err( *Extended ) + Access( *NoLock )
Key: Feld oder KeyList
Err: *INxx oder *Extended für Abfrage über %Error
Access: *NoLock für keine Sperre
```

Wie auf i5/RPG wird zum aktualisieren der **Befehl Update** verwendet.

```
Update
File( CMMastF ) +
Err( *IN89 )
```

Update unterstützt wie auf i5/RPG Record-Formatnamen für Multiformat-Logical-Files.

Der Rucksack der Vergangenheit

Die größten Hindernisse bei einer Modernisierung sind jene Eigenheiten, die jahrzehntelang als die ‚Stärken‘ der i5 gepriesen wurden. Dinge wie Teildateien, MatchingRecord oder der Nachfolger MultiformatLogicalFiles sind nur auf der i5 zu finden. So praktisch sie auch sind, wurden sie zum Sargnagel für die Software, die sie einsetzt. Da es dafür keine Entsprechung auf anderen Systemen gibt, ist eine Code-Migration von RPG in Sprachen wie VB, C# oder Java unmöglich.

Mit VisualRPG können diese Tabellen problemfrei angesprochen werden; es braucht an der Logik – außer der Syntax – auch nichts geändert werden.

Dass man sich von diesen Technologien langfristig trennen muss, ist klar. Es ist aber eine unschätzbare Hilfestellung, wenn sie unterstützt werden, da sehr viel Druck aus dem Erneuerungsprozess genommen wird.

In der Geschichte der i5 hat es immer wieder Migrationsangebote gegeben. Die waren meist so ausgerichtet, dass auf einer Plattform ein iSeries-ähnliches Umfeld aufgebaut wurde, von dem man nur sehr schwer wieder weggekommen ist. Eine Integration von RPG und Windows ist hingegen nur mit ASNA möglich. Neu ist auch, dass mit .Net eine Entwicklungsplattform zur Verfügung steht, die weltweit akzeptiert ist und alle bekannten Sprachen vereint. Somit ist auch die Frage des Generationswechsels der Entwickler entschärft.

Wege zur Erneuerung

Mit VisualRPG stehen viele Wege offen. ASNA bietet ein Migrations-Tool von i5 auf .Net an.

MONARCH analysiert die i5-Objekte, sucht sich die Sourcen und erstellt daraus ASPX-Panels, VisualRPG-Klassen, PrintFiles und Project-Solutions in VisualRPG.Net.

Aus den bereits genannten Gründen ist eine PushButtonSolution nicht möglich. Es gibt schlicht und ergreifend keinen Weg, RPG-Code ohne manuellen Eingriff auf .Net zum Laufen zu bringen.

Die Höhe des zu leistenden manuellen Einsatzes hängt sehr stark vom i5-Code ab. Business-Logik wird problemlos migriert. Aufzulösen sind überlappende Datenstrukturen mit gepackten Feldern etc. Der größte Aufwand ist üblicherweise im Dialogbereich zu leisten. Wurde hier gezielt am 5250-Konzept vorbeiprogrammiert, so ist eine Migration schwieriger als wenn übliche Subfiles und Datenfelder verwendet werden.

Sehr oft ist es aber gar nicht nötig eine Anwendung komplett zu migrieren, um spürbare Verbesserungen zu erreichen.

In vielen Fällen hilft es schon sehr, wenn Schlüsselprogramme oder Schnittstellen zum System erneuert werden. Dies kann von einem RPG-Programmierer auch in wenigen Wochen erreicht werden.

Ein idealer Einstieg sind die bei der Installation mitgelieferten

Anwendungsbeispiele für Windows, Web und Web-Services. Nach der Installation kann man VisualRPG 30 Tage kostenlos testen. Während dieser Zeit sollte man sich die Beispiele ansehen und als ersten Schritt versuchen aus den Beispielen auf eigene Tabellen zuzugreifen. Meistens treten bereits an diesem Punkt die in VisualRPG enthaltenen Möglichkeiten für das eigene Umfeld hervor.

ASNA und IBM

Diese beiden Unternehmen verbindet eine langjährige Zusammenarbeit. ASNAs Techniker erstellten Anfang der 90er Jahre im Auftrag von IBM einen C-Compiler, der im Bereich der Datenbank verwendet wurde. Davor bot ASNA einen Debugger und eine Datenbank auf den Systemen S/3x an. 1995 stellte ASNA

VisualRPG 1.0 vor und war somit der Vorreiter der Integration von i5-Daten auf Windows und Web.

Fazit

VisualRPG.Net ist eine ideale Möglichkeit, Investments in die i5-Welt in eine zeitgemäße IT-Landschaft zu retten. Durch die Abstraktion des RPG-Codes von der Hardware-Plattform gewinnt der Anwender viele neue Möglichkeiten. Er kann die i5 weiter verwenden und seine Anwendungen schrittweise modernisieren. Besonders interessant ist dieses Angebot für Software-Häuser, die ihre Anwendungen auf eine neue Plattform bringen müssen. Für Unternehmen, die i5 einsetzen, bietet VisualRPG die Möglichkeit, die Anwendungen mitsamt der Betreuungsmannschaft auf die .Net-Plattform überzusiedeln.

Für Green-Screen-Programmierer ist VisualRPG eine Herausforderung. Die Schreibweise der Befehle (OpCodes) ist vertraut; die Umgebung empfindet man im der Anfangsphase gewöhnungsbedürftig und man braucht einige Zeit, um sich darauf einzustellen.

Die Midrange-Welt kann sich dem angestauten Erneuerungsbedarf nicht verschließen. Vor einigen Jahren wurden Programme zwar mit Webfacing versehen, unter der hübschen Maske blieb aber alles beim Alten und eine echte Erneuerung wurde dadurch nur auf unbestimmte Zeit verschoben.

Nicht zu unterschätzender Druck kommt auch aus dem Umfeld, wenn Funktionen eingefordert werden, die erst unter .Net einfach umzusetzen sind. Dazu gehören zum Beispiel die B2B-Kommunikation, das Anbieten und Nutzen von Web-Services oder der Datenaustausch mit Office-Anwendungen. Grundsätzlich ist dies auch unter i5/OS möglich. Einfacher, moderner und somit kostengünstiger geht's aber mit VisualRPG.Net.

Fachautor: Christian Neißl