

Die IBM Rational Migration Extension for System i (RMEi) Eine zukunftsweisende Lösung

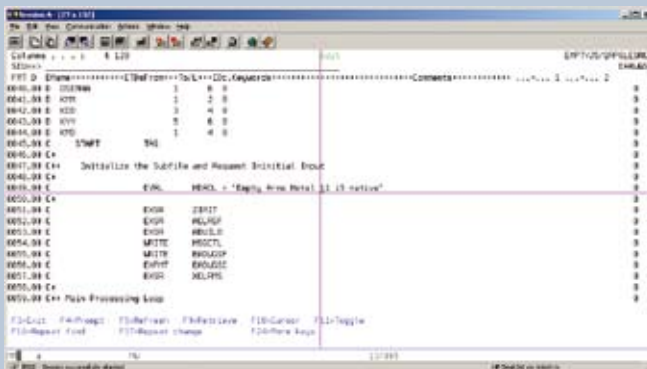
Der Paradigmen-Bruch zwischen traditioneller prozeduraler Programmierung und objektorientierter Welt erregt auch im Krisenjahr 2009 die Gemüter der Entwickler und Software-Architekten. Die Kernfrage, um die es dabei geht: Wie können die Stärken beider Ansätze (schnell und produktiv entwickeln sowie modernste Technologie für den Anwender anbieten) sinnvoll kombiniert werden?

Auf diese Frage gibt es genau eine Antwort: Umsteigen auf die IBM Business-Sprache EGL (Enterprise Generation Language). EGL ist eine Programmiersprache der neuesten Generation, die sowohl den client- als auch den serverseitigen Bedarf bei der Entwicklung von Geschäftsanwendungen

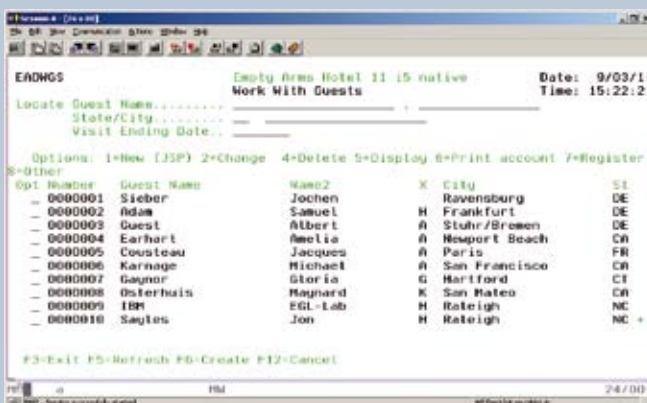
berücksichtigt: Moderne Frontends, basierend auf JSF und Ajax-Technologie, lassen sich ebenso herstellen wie klassische Batch-Programme mit Text-User-Interface à la 5250. Das Schöne daran: Aus EGL heraus kann man direkt Java-Programme erzeugen, aber weiterhin auch native i-Programme. Doch ganz

besonders wichtig: EGL ist keine objektorientierte, sondern eine prozedurale Programmiersprache, in der die modernen SOA-Aspekte implizit eingebunden sind. Die bisher fehlende tragfähige Brücke zwischen traditioneller und objektorientierter Welt ist somit gebaut. Die EGL Entwicklungsumgebung ist in

1 VERFOLGUNG DER ÄNDERUNGEN



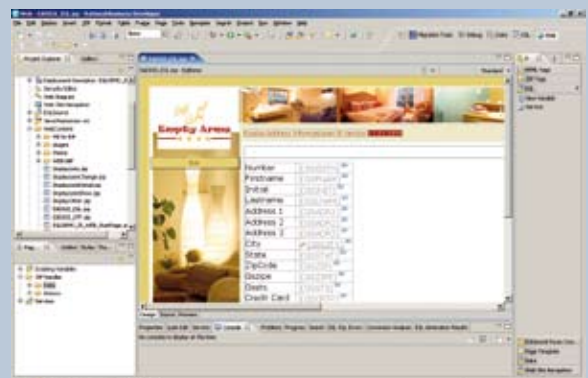
2 5250-ANSICHT



3 RMEI-ANSICHT



4 BILDSCHIRMGESTALTUNG IN EGL



folgenden Paketierungen erhältlich:

- IBM Rational Business Developer (RBD)
- IBM Rational Developer for System i for SOA Construction (RDi for SOA)

Soweit, so gut. Doch wie lassen sich bewährte RPG-Anwendungen sicher und effizient nach EGL migrieren, um nachhaltig auch deren Return of Invest zu sichern?

Hierfür bietet IBM die Lösung mit dem Produkt:

- IBM Rational Migration Extension for System i (RMEi)

Mit dem RMEi können die vorhandenen RPG-Programme nach EGL migriert werden. Dazu stellt der RMEi zwei Optionen zur Verfügung:

- RMEi Web-API: erzeugt aus den interaktiven Programmteilen ad hoc eine XML-basierte Browser-Oberfläche und ermöglicht so die nahtlose Integration mit EGL-Neuentwicklungen
- RMEi Konverter: transformiert RPG-Code in EGL-Code

Der große Vorteil beim Umstieg auf EGL ist: Kein Paradigmenbruch mehr zwischen prozeduraler und objektorientierter Welt. Dieser überforderte die meisten Anwendungsentwickler und führte in der Praxis zu schwer wartbaren Ergebnissen. Eine Modernisierung der Benutzeroberfläche, wie von zahlreichen Kunden in den letzten Jahren durchgeführt, ist zwar ein guter Anfang

- doch reines Screenscraping allein erhöht nur die Komplexität, ohne die Funktionalität nachhaltig zu verbessern!

Wann ist eine Umstellung auf EGL sinnvoll?

- Wenn die Grundsatzdiskussion aufkommt, ob man die Eigenentwicklung durch Standardsoftware ablösen soll.
 - Häufig der Fall, wenn neue Manager in der IT oder in der Geschäftsleitung aktiv werden.
- Wenn neue Anforderungen nicht mehr in RPG realisiert werden, sondern unter Java oder .NET.

■ Wenn es eine größere Zahl von Anwendern gibt, die ständig zwischen GUI und Green Screen Oberflächen verschiedener Anwendungen hin- und herwechseln müssen.

■ Wenn Programmanpassungen und Erweiterungen nicht mehr zeitnah ausgeführt werden können - und die Entwicklung zum erfolgskritischen Flaschenhals wird.

Im vorliegenden Beitrag wollen wir uns Schritt für Schritt in vereinfachter Form anschauen, wie eine solche sanfte Migration funktioniert.

Betrachten wir also zunächst ein typisches RPG (RPG IV) Programm (Abb. 1). Die rote Markierung im Bild erleichtert später die Verfolgung der Änderungen.

Wenn wir das Programm kompilieren und dann wie gewohnt als 5250 Anwendung starten erhalten wir den Screen aus Abb. 2.

Nun nutzen wir die RMEi Web-API und starten die gleiche Anwendung (vgl. Abb. 3).

Wir erhalten eine moderne GUI, die alle Möglichkeiten der Integration von „neuer“ Technologie, z. B. EGL JSP Seiten, in die bestehende RPG Anwendung ermöglicht. Technisch detaillierter wird die RMEi Web-API an einer späteren Stelle in diesem Artikel beschrieben.

Soweit, so gut. Schauen wir uns nun die EGL Entwicklungsumgebung an, hier als Beispiel eine Bildschirmgestaltung in EGL einer Detail-Anzeige zu obiger Übersichts-Maske (diese basiert ja noch auf unserem Ursprungs-RPG-Source).

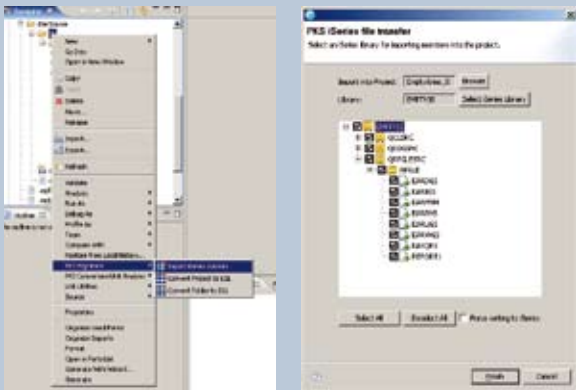
Markiert wird hier das Datum (11.03.2009), auf das wir später zurückkommen werden (Abb. 4).

Über das Kontext-Menü unserer Übersichts-Maske können wir nun auf die neu erstellte JSP Seite verzweigen. Innerhalb dieser EGL JSP Seite wird ein WEB-Service aufgerufen, welcher das aktuelle Wetter der angezeigten Stadt ermittelt (vgl. Abb. 5).

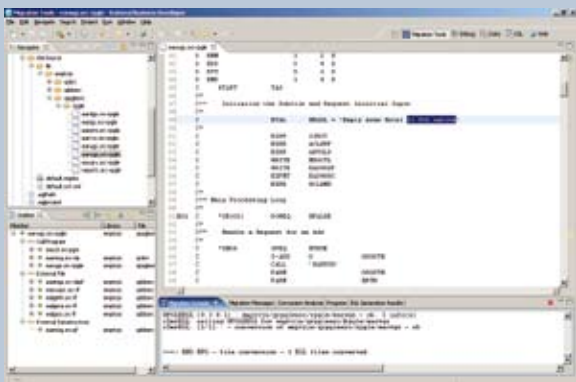
5 EGL JSP SEITE MIT WEB-SERVICE



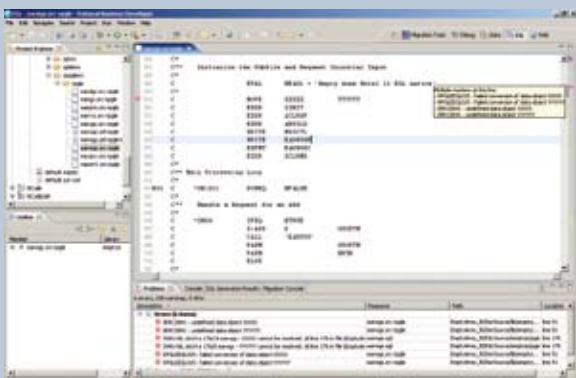
6 IMPORT WIZARD



7 MIGRATION



8 PROBLEMAN SICHT BEI FEHLERN



alle verwendeten PF, LF, DSPF und PRTF Sourcen inklusive Referenzfiles.

Über einen Import Wizard werden die entsprechenden Sourcen aus den System i Members in eine geeignete EGL Projektstruktur übertragen (vgl. Abb. 6).

Alle Sourcen werden in eine standardisierte Projektstruktur übertragen. Hierbei wird für jede Library ein eigenes Verzeichnis angelegt und entsprechende Unterverzeichnisse für Sourcefiles und Sourcemembers inklusive der verschiedenen Sourcearten.

Nun erfolgt ein Wechsel der Perspektive auf die „RPG – EGL Migration“. In dieser Perspektive finden sich alle für die Migration benötigten Werkzeuge.

2. Die Migration kann nun gestartet werden. Sie kann erfolgen Source für Source oder als Gesamtmigration für ein komplettes Projekt- bzw. Teilprojekt.

Während die Migration läuft, werden auf der „Migration Tooling Console“ alle ausgeführten Schritte angezeigt. Nach Abschluss der Migration erfolgt auf der Konsole eine Meldung über die Anzahl der migrierten Programme.

Hier wird nun die am Anfang des Artikel geänderte RPG Source nochmals geändert in „11 EGL native“ (Abb. 7).

Um zu sehen, was passiert, wenn in der Migration der Programme ein Fehler auftritt, simulieren wir ein Problem mit einem MOVE Befehl im Programm.

Der Befehl MOVE XXXX YYYYYY arbeitet mit nicht deklarierten Feldern XXXX und YYYYYY (Abb. 8).

Karlheinz Peter ■

So leicht und flüssig können also vorhandene RPG-Anwendungen mit der RMEi Web-API in ein Browser-Frontend überführt und nahtlos mit Neu-Entwicklungen in JSP-Technologie eingebunden werden.

Doch nun zur zweiten wichtigen Option der RMEi: dem Konverter zur Migration von RPG nach EGL. Dazu

beschreiben wir nachfolgend die einzelnen Schritte, die nötig sind, um existierende RPG-Anwendungen nach EGL zu überführen:

1. Zunächst werden die existierenden Sourcen von der iSeries in die EGL Rational Entwicklungsumgebung überführt. Hierbei werden nicht nur die RPGLE Sourcen benötigt, sondern auch

CLICK TO

www.midrange.de

Den vollständigen Artikel finden Sie in unserer aktuellen Online-Ausgabe.