

RULAMAN® UND SPL

DIE INTEGRIERTE MIGRATIONSUMGEBUNG

Auf einen Blick

RULAMAN ist die generalisierte Plattform für die automatisierte Konversion von Quellcode. Integraler Bestandteil ist die speziell für dieses Umfeld entwickelte *Source Code Processing Language* SPL, in der Algorithmen zur Transformation formuliert werden. RULAMAN dient als Arbeitsumgebung sowohl für den Regelentwickler als auch für den Migrator, der Konversionsprozesse beliebiger Komplexität automatisiert ablaufen lässt.

DIE INTEGRIERTE MIGRATIONSUMGEBUNG

RULAMAN ist die Plattform für die Transformation von Quellcode in beliebigen Ausgangs- und Zielsprachen. Sie enthält Mechanismen für die Kompilation und Ausführung von in SPL geschriebenen Regeln.

Die zu verarbeitenden Entitäten sind ebenso wie die für den Ablauf erforderlichen Parameter in einem eigenen Repository gespeichert. Dort werden auch die Ergebnisse der Transformation, weitere sekundäre Daten und Log-Dateien abgelegt.

Regeln können einzeln, in Gruppen oder als vollständige Regelsätze, sog. *Rule Sets* ausgeführt werden. Im Produktionsmodus verarbeitet RULAMAN beliebig große Stapel von Entitäten auf einmal.

PARSER UND EXPORTER

Das Produkt verfügt über eine große Zahl von *Parsern*, u. a. für COBOL, Java, CSP, Job Control oder Informix 4GL. Dazu kommen Listparser für eine Reihe von Mainframeassemblern.

Exporter lesen das Ergebnis der Transformation aus dem RULAMAN Repository und geben es als neuen Quellcode in der sprachspezifischen Form aus.

REGELN, REGELSÄTZE UND REGELSTEUERUNG

Regeln und Regelsätze können auf sehr flexible Art in hierarchischen Strukturen angeordnet werden. Somit ist auch die Regelsteuerung über SPL möglich.

Regelsätze für häufig vorkommende Aufgaben wie Assembler nach COBOL, CSP nach Java oder PL/I nach COBOL sind als eigene Produkte ausgeprägt.

DIE MODERNISATION FACTORY

Wir verbinden bestehende Konversionsprodukte mit neuen oder kundenspezifisch erweiterten Konvertern zu einer sog. *Modernisation Factory*, ähnlich einer industriellen Fertigungsstraße.

Mit der so entstandenen *Factory* können beliebige Mengen von Quellcode und die zugehörigen Daten vollautomatisch innerhalb von Stunden vollautomatisch in die Zielumgebung überführt werden.

HIGHLIGHTS

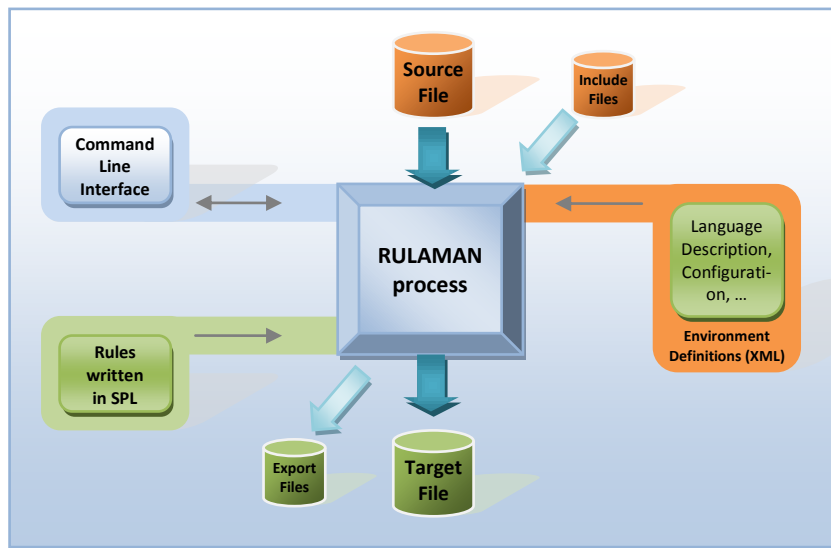
- › erlaubt die extrem schnelle Entwicklung von Konvertern
- › dient dem Konverter-Entwickler und den Migrator
- › hohe Performance dank Rule-Compile in Java Byte Code
- › SPL zur Realisierung von Rules und Ablaufsteuerung
- › Regelsätze leicht anzupassen und zu kombinieren
- › erlaubt die Konversion von jeder Sprache in jede andere
- › viele unterschiedliche Parser und Exporter verfügbar
- › SPL ist leicht zu lernen

SPL UND LANGUAGE PACKS

Wir haben SPL speziell für die automatisierte Transformation von Quellcode entwickelt. Alle unsere Konversionsprodukte sind in SPL geschrieben.

SPL ist eine *domain specific language* mit einer sog. adaptiven Grammatik. Das bedeutet, dass sich die SPL Sprachsyntax an Umgebungsbedingungen anpasst. Beispiel: bei SOURCE = ASSEMBLER kann das Keyword OPCODE genutzt werden, andernfalls wird OPCODE als Fehler ausgewiesen.

Eine weitere Besonderheit sind die *Language Packs*. Sie spezifizieren als externe XML-Dateien die Syntax der Statements von Ausgangs- und Zielsprachen. So kann RULAMAN und seinem eingebauten Compiler auf einfache Weise „von außen“ die jeweilige Sprachsyntax bekannt gemacht werden.



- Lokale und globale Prozeduren sind ebenso verfügbar wie lokale und globale Variablen, inklusive der sog. Positionsvariablen, die eine leichte Navigation im Quellcode erlauben
- Neben den Ein- und Ausgaben für Quellcode sind sekundäre Dateien möglich, über die z.B. COPY Entitäten eingelesen werden oder CSV- und Log-Dateien ausgegeben werden können.
- Spezielle Aufgaben, die den Rahmen der SPL sprengen würden, sind in Java-Service-Routinen ausgelagert.

➤ *Pattern Matching* erlaubt das schnelle und einfache Auffinden bestimmter Codemuster, z.B. „finde eine 3-fach geschachtelte IF-Klausel“.

➤ Der Einsatz mehrerer Parser

pro Entität erlaubt die Verarbeitung gemischt-sprachiger Entitäten, z.B. *embedded SQL* oder *Columbus* in COBOL.

DER SPL INSTRUKTIONSSATZ

Entwickelt für die Sprachtransformation enthält SPL viele Befehle, die sich mit Statements, Tokens, Keywords, Codeblöcken und Codefragmenten befassen.

- Jede Regel hat eine festgelegte Grundstruktur, die RULAMAN u.a. mitteilt, unter welchen äußeren Bedingungen die Regel zu aktivieren ist.
- SPL bietet die für prozedurale Sprachen üblichen Elemente zur Steuerung: Bedingungen, Schleifen usw.

TECHNOLOGIE

Durch die Implementierung in Java ist RULAMAN unabhängig vom Betriebssystem: Er kann auf jeder Plattform ablaufen, die eine *Java Virtual Machine* besitzt, z.B. Linux, Unix oder Windows.

Der RULAMAN Workload kann durch einen eingebauten Dispatcher auf beliebig viele Workstations oder Server verteilt werden.