

Arlanis Universal Data Converter 3.0

Neue Wege zur Datenintegration

Die moderne Form der Datenintegration stellt neue und vielfältige Anforderungen an heutige Werkzeuge und deren Funktionalität und Handhabung. **Andreas Holubek, Joerg Selig**

Auf einen Blick

Inhalt

Eine gute Unterstützung durch geeignete Werkzeuge ist bei Datenintegrationsprojekten unter dem Aspekt der zunehmenden Komplexität von Anwendungsarchitekturen besonders wichtig. Dieser Artikel gibt eine kurze Einführung in die Datenintegration und stellt eine moderne Vorgehensweise vor.

Technik

Datenmigration, Datenintegration, Datenkonvertierung

Voraussetzung

arlanis Universal Data Converter 3.0

Die Autoren

Andreas Holubek arbeitet als VP Engineering für die arlanis Software AG. Sein besonderes Interesse gilt SOA, Web Services, EAI und den verschiedenen Persistenztechniken. Er ist bekannt als Autor und Referent auf verschiedenen Fachkonferenzen.

Joerg Selig ist als Softwareentwickler für die Arlanis Software AG tätig. Seine Schwerpunkte sind Java, Eclipse-Technologien und das Design von Benutzerschnittstellen.

Der Begriff Datenintegration wird oft in den Zusammenhang mit Datenmigration und Datenkonvertierung gestellt. Für alle drei Begriffe gibt es keine feste, einheitliche Definition. Es hat sich jedoch in der IT-Welt ein Konsens gebildet, was man im Kern darunter zu verstehen hat. Ganz allgemein gesprochen geht es in erster Linie um den Transfer von Daten. Sei es der Transfer von einem Datenformat in ein anderes, der Transfer zwischen verschiedenen Speichermedien, zwischen Anwendungen oder zwischen Computersystemen. Interessant sind die Fälle, bei denen sich zwischen Quelle und Ziel das Datenformat derart unterscheidet, dass es eine strukturelle Differenz zu überbrücken gilt. In einigen Publikationen finden Sie den Begriff der Datenmigration als Oberbegriff für Datenkonvertierung und Datenintegration. An anderen Stellen wird zwischen Migration und Integration unterschieden, und der Begriff der Konvertierung findet keine Verwendung. Wir wollen uns der zweiten Sicht anschließen und zusätzlich die Datenkonvertierung als den Teil der beiden Konzepte betrachten, bei dem es um den Transfer von Daten zwischen verschiedenen Formaten und Strukturen geht.

Daten migrieren und integrieren

Eine Datenmigration geht oft mit einer Softwaremigration einher. Hierbei wird eine eingesetzte Software durch eine neue abgelöst. Das kann Software eines anderen Herstellers sein oder nur eine neue Version, wobei im zweiten Fall eine Datenmigration eher selten ist. Bei einer Datenmigration besteht die Anforderung, die Daten aus dem Altsystem in die neue Soft-

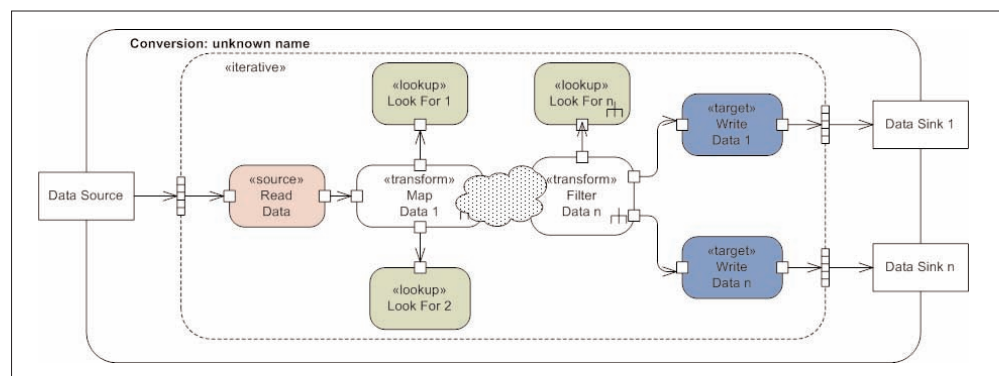
ware zu überführen. Wenn alte und neue Software unterschiedliche Datenformate oder Datenstrukturen nutzen und die Export- und Importfunktionen diese Lücke nicht überbrücken, wird eine explizite Datenkonvertierung notwendig. Die Datenmigration ist ein einmaliger Vorgang.

Im Gegensatz zur Datenmigration ist die Datenintegration ein permanenter Prozess, bei dem es um den Austausch von Daten zwischen verschiedenen Anwendungen beziehungsweise Datenspeichern in einer komplexen IT-Architektur geht. Damit steht die Datenintegration in engem Zusammenhang mit der Software-beziehungsweise Anwendungsintegration. Die Datenintegration kann im Wesentlichen unter zwei Aspekten betrachtet werden. Der erste Aspekt betrifft die Konvertierung von Daten, während es beim zweiten Aspekt um die integrativen Anforderungen geht. Da die Datenintegration ein permanenter Prozess ist, geht es beim integrativen Aspekt hauptsächlich um die Steuerung und Überwachung der Datenkonvertierung. Die Frage, was die Datenintegration ist, kann also mit „Datenkonvertierung plus zusätzlicher integrativer Eigenschaften“ beantwortet werden. Wenn wir von Datenkonvertierung reden, meinen wir immer den Aspekt der Datenintegration, der den Transfer von Daten betrifft. Das heißt, wir sehen die Datenkonvertierung immer im Kontext der Datenintegration.

Integration as a Service

Die Datenintegration ist fester Bestandteil in etablierten Integrationsarchitekturen. Zum Beispiel gibt es in der ESB-Architektur (Enterprise

Schematische Darstellung einer Konvertierung nach dem datengraph- und datensatzbasierten Paradigma (Bild 1)



Service Bus) einen speziellen Integrationsdienst, der für die Überbrückung von verschiedenen Datenformaten und Datenstrukturen zuständig ist. Es ist zu beobachten, dass Datenintegration zunehmend im Zusammenhang mit Integrationsarchitekturen anzutreffen ist. Stellt man Datenintegration in diesen Kontext, kommt man fast zwangsläufig auf ein Konzept, das wir mit Integration as a Service (IaaS) bezeichnen. Die integrativen Eigenschaften werden explizit zu Diensteseigenschaften, sodass die Datenintegrationslösung direkt als Integrationsdienst verwendet werden kann. Sei es innerhalb eines ESB oder als Dienst, der allgemein über eine Webdienstschnittstelle angeboten wird.

Daten konvertieren

Die Datenkonvertierung, kurz Konvertierung, ist elementarer Bestandteil der Datenintegration und befasst sich mit dem Transfer von Daten. Dabei müssen sowohl verschiedene Datenformate als auch Datenstrukturen aufeinander abgebildet werden. Zwischen dem Lesen und

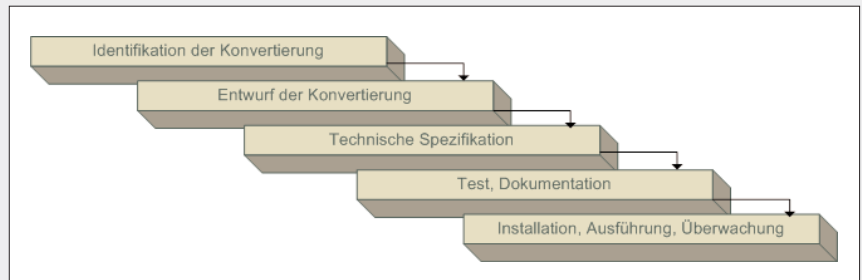
Schreiben der Daten aus beziehungsweise in die Datenformate sind 0...n Transformationen dieser Daten vorzunehmen. Dabei wird ein Datensatz in einen anderen überführt. Der Vorgang einer Transformation kann verlustbehaftet sein, das heißt, nicht alle Informationen aus dem Quelldatensatz können zwangsläufig in den Zieldatensatz überführt werden. Zum anderen kann der Zieldatensatz mit weiteren Informationen aus anderen Datenquellen angereichert werden, was landläufig mit Lookup bezeichnet wird. Ein Lookup beschreibt eine Schlüsselbeziehung zum aktuell bearbeiteten Datensatz. Er liefert alle Datensätze aus einer Lookup-Datenquelle, für die eine bestimmte Bedingung erfüllt ist. Im einfachsten Fall ist dies der Wert eines Feldes, der mit dem Wert eines Feldes aus dem aktuellen Datensatz übereinstimmt.

Paradigmen für die Konvertierung

Eine generelle Vorgehensweise während der Abarbeitung einer Konvertierung kann mit einer

Vorgehensweise

Das grundlegende Vorgehen bei einem Datenintegrationsprojekt kann verallgemeinert werden. Diese Vorgehensweise wollen wir hier nur grob in fünf Phasen skizzieren (Bild 2), sie kann jedoch noch weiter verfeinert werden, sodass am Ende eine detaillierte Schritt-für-Schritt-Anleitung steht, die ein ingenieurmäßiges Arbeiten zulässt.



Vorgehen in fünf Phasen (Bild 2)

- **Identifikation der Konvertierung:** Hier ist zu klären, welche Daten von wo nach wo zu konvertieren sind. Die Fragen, die sich dabei stellen, lauten: Aus welchen Quellen sollen die Daten gelesen werden? Welche Quellen sollen als Lookup-Quellen dienen? In welche Senken sollen die Daten geschrieben werden? Durch die Identifikation der Datenquellen und -senken haben Sie inhärent die Frage nach den beteiligten Datenformaten geklärt.
- **Entwurf der Konvertierung:** Folgt das eingesetzte Werkzeug dem Datengraph-Paradigma, besteht die Spezifikation der Konvertierung aus der Identifikation der notwendigen Schritte einer Konvertierung sowie deren Beziehungen untereinander. Die erste Phase liefert bereits die Schritte für das Lesen beziehungsweise Schreiben der Daten. Die weiteren Schritte sind Transformationen, die zwischen den Quellen und Senken liegen und mit ihnen verbunden sind. Diese Transformationen können kaskadiert, also ebenfalls miteinander verbunden werden. Für die Identifikation der Transforma-

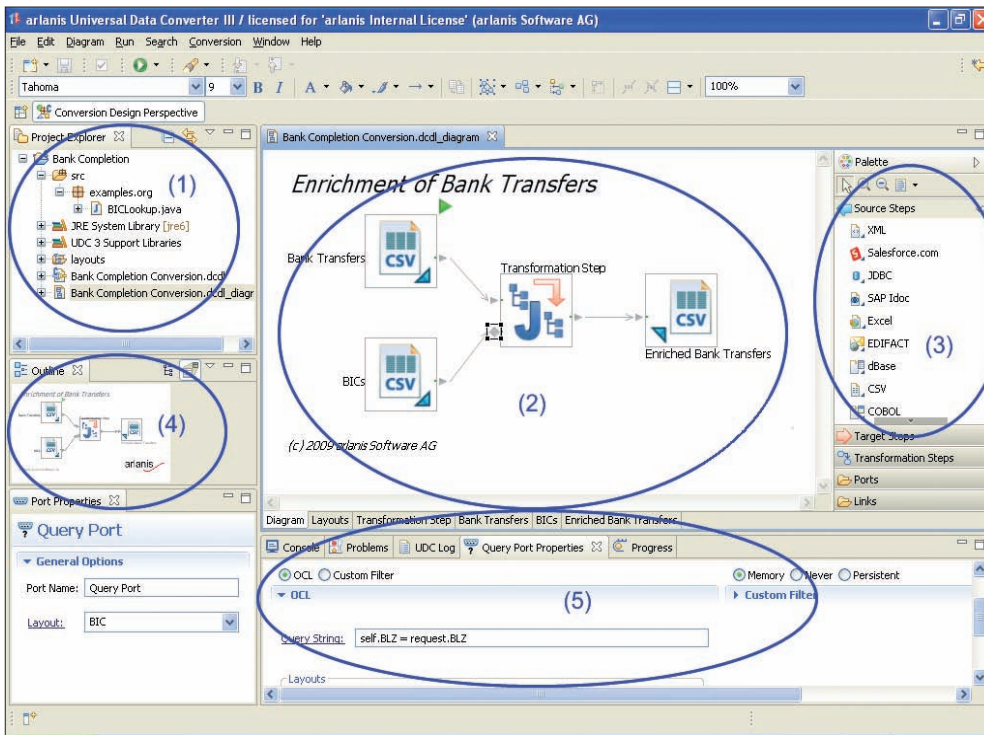
tionen benötigen Sie Wissen über die Anwendungsdomäne, in deren Kontext die Konvertierung stattfindet. Wollen Sie zum Beispiel Kundendaten aus einer Datenbank unter Anreicherung von Daten aus einer zweiten Datenbank in eine Zieldatenbank transferieren, wird eine Transformation genügen, sodass Sie insgesamt vier Schritte identifiziert haben, die über drei Kanten miteinander verbunden sind.

- **Technische Spezifikation:** Bei der technischen Spezifikation wird die Detailtiefe erhöht, in der die einzelnen Schritte parametrisiert werden. Bei den Quellen und Senken sind das einmal die formatspezifischen Angaben, die zum Lesen und Schreiben benötigt werden, wie zum Beispiel Dateinamen oder Zugangsdaten für Datenbanken. Als Nächstes müssen die Datenstrukturen definiert werden, mit denen die von den Quellen emittierten beziehungsweise von den Senken konsumierten Datensätze beschrieben werden. Durch die Verbindung der

Transformationen mit anderen Schritten ist indirekt klar, welche Strukturen innerhalb der Transformation aufeinander abzubilden sind. Es bleibt hier die Aufgabe, diese Abbildung zu definieren. Im einfachsten Fall reicht es aus, einzelne Felder eines Datensatzes abzubilden.

- **Test und Dokumentation:** Dass eine Konvertierung fehlerfrei funktionieren soll und daher einem oder mehreren Tests zu unterziehen ist, liegt auf der Hand. Ein nicht zu unterschätzender Punkt besteht in der Dokumentation einer Konvertierung. Diese erhöht die Wartbarkeit sowie das Lokalisieren von Fehlern, die nicht durch Tests aufspürbar sind.
- **Installation und Ausführung:** Der letzte Punkt besteht aus der Installation einer Konvertierung in der Ausführungsumgebung sowie deren anschließender Ausführung und Überwachung. Sollten hierbei Probleme auftreten, kann es zu weiteren Iterationen über die hier beschriebenen fünf Punkte kommen.

Die einzelnen Bestandteile des UDC-Designers (Bild 3)



Anlehnung an den ETL-Prozess beschrieben werden, der im Umfeld des Data Warehousing definiert wurde. ETL steht für „Extract“, „Transform“, „Load“. „Extract“ steht für das Extrahieren der Daten aus den Quellen. Dabei müssen die extrahierten Daten nicht vollständig sein, das heißt, bereits bei der Extraktion kann eine Vorfilterung erfolgen. „Transform“ ist derjenige Teil des Prozesses, bei dem ein oder mehrere Quelldatensätze in einen oder mehrere Zieldatensätze überführt werden. Und „Load“ steht für das Laden der transformierten Daten in die Datensinken. Für die Umsetzung eines derartigen Prozesses sind viele Ansätze denkbar. Für einfache und einmalige Konvertierungen ist es ausreichend, ein kleines Programm zu schreiben, das die Konvertierung übernimmt. Für den Zugriff auf die Daten und andere Aufgaben können Bibliotheken herangezogen werden. Die Umsetzung des ETL-Prozesses ist dann Implementationsdetail.

Für komplexere Konvertierungen empfiehlt sich in der Regel der Einsatz einer speziellen Software, die Entwicklung und Ausführung von Konvertierungen unterstützt. Wie eine Software den oben beschriebenen Prozess umsetzt, ist eine Frage des dieser Anwendung zugrunde liegenden Paradigmas. Ein mögliches Paradigma besteht in der Komposition von Konvertierungen aus einzelnen dedizierten Schritten. Derartige Schritte sind zum Beispiel das Lesen aus einer Datenquelle, das Schreiben in eine Datensinke oder eine einzelne Transformation. Bei dieser Architektur kann man eine Konvertierung als einen einfachen, gerichteten Graphen (Datengraph) auffassen. Die Menge der Knoten des Graphen besteht aus den einzelnen Schritten der Konvertierung. Eine gerichtete Kante steht dafür, dass ein Schritt dem folgenden Schritt Datensätze liefert. Die Daten der Quellen werden Datensatz für Datensatz ▶

DCDL

Die vollständige Konvertierung ist mit der domänenspezifischen Sprache DCDL beschrieben. DCDL steht für Data Conversion Description Language und wird in XML abgelegt. Der UDC interpretiert die DCDL und erzeugt zur Laufzeit ein ausführbares Artefakt der Konvertierung.

Tabelle 1: Bestandteile des Universal Data Converter in der Übersicht

Komponente	Beschreibung
UDC-Designer	Hierunter ist das Cockpit zu verstehen, in welchem die Konvertierungen erstellt, getestet und für die Auslieferung verpackt werden. Hier finden Sie genau jenen Bestandteil, den Sie in der Rolle des Entwicklers benötigen. Der Designer enthält eine Stand-alone-UDC-Runtime, sodass für das Ausprobieren keine Plattform erforderlich ist.
UDC-Plattform	In der Plattform leben die fertiggestellten Konvertierungen. Sie können nach Bedarf gestartet, planmäßig ausgeführt und beendet werden. Die Plattform verfügt über eine Web-Service-Schnittstelle für das Management und Monitoring der Konvertierungen und passt sich perfekt in die Welt des Cloud Computing ein.
UDC-Monitor und Management-Konsole	Mithilfe dieser Konsole bekommen Sie einen ansprechenden grafischen Überblick über das, was in der Plattform gerade passiert. Also kurz gesagt: Welche Konvertierung ist in welchem Zustand?
UDC-Runtime-Engine	Der Vollständigkeit halber sollte noch erwähnt werden, dass sich eine Konvertierung auch ohne Plattform und Designer ausführen lässt. Dafür ist die UDC-Runtime-Engine hilfreich. Allerdings hat man hier nicht so schöne grafische Administrationswerkzeuge.

Basisdatenformate

- CSV, Text
- COBOL
- Custom Code
- Microsoft Excel
- XML
- RDBMS, JDBC, ODBC
- dBase
- EDIFACT
- SAP iDoc

Module

- Grafische Transformation
- Java und Scripting
- Conversion Call
- Synchronize
- Source-Adapter
- Target-Adapter
- Source- und Target-Adapter jeweils für alle Datenformate

durch den Graphen geleitet und transformiert. Dieses Paradigma nennen wir datengraph- und datensatzbasiert. Bild 1 veranschaulicht dieses Paradigma exemplarisch.

Anforderungen an Werkzeuge für die Datenintegration

Aus dem vorigen Abschnitt, aber auch aus eigener Projekterfahrung lassen sich Anforderungen an Werkzeuge definieren, die den Anwender bei der Durchführung von Datenintegrationsprojekten unterstützen. Eine Integrationslösung sollte grundsätzlich auch für die Abwicklung von Datenmigrationsprojekten geeignet sein, das heißt, sie muss es auf einfache Weise erlauben, eine Konvertierung auszuführen, ohne dass die integrativen Eigenschaften zum Tragen kommen.

■ **Große Vielfalt an Formaten:** Ein Werkzeug sollte möglichst universellen Charakter haben und viele Datenformate unterstützen. So können Sie schnell auf neue Anforderungen im Projekt reagieren.

■ **Design einer Konvertierung auf hohem abstrakten Niveau:** Die Möglichkeit des Designs einer Konvertierung auf einem hohen abstrakten Niveau ist in vielen Phasen des Lebenszyklus einer Konvertierung hilfreich. Das oben beschriebene datengraph- und datensatzbasierte Paradigma eignet sich sehr gut für diese Anforderung, da sich Graphen sehr gut visualisieren lassen.

■ **Gute visuelle Unterstützung bei großer Detailtiefe:** Auch wenn das grobe Design einer Konvertierung noch so einfach und komfortabel ist, irgendwann muss der Designer einer Konvertierung die Details spezifizieren. Hier sollte ein Werkzeug den Anwender auch bei den sehr technischen Details unterstützen und nicht absolutes Fachwissen voraussetzen.

■ **Nachvollziehbarkeit:** Die Nachvollziehbarkeit des Ablaufs von Konvertierungen muss gewährleistet werden, und zwar auf der Ebene des Lebenszyklus. Hier interessieren Fragen wie: Ist die Konvertierung gelaufen? War sie ohne Fehler? Wann und wie lange ist sie gelaufen? Besonders bei verlustbehafteten Konvertierungen ist die gute Nachvollziehbarkeit des Ablaufs von Konvertierungen wichtig. Also wird ein detailliertes Protokoll über gelesene und geschriebene Datensätze gefordert.

■ **Ausführungsumgebung:** Eine Konvertierung mit einem Werkzeug zu entwerfen und in der Entwicklungsumgebung testweise laufen zu lassen, ist eine Sache, sie dann in einer produktiven Umgebung einzusetzen, eine andere. Eine Ausführungsumgebung, die den Gegebenheiten einer produktiven Umgebung gerecht wird, muss einige Anforderungen erfüllen, besonders unter dem Gesichtspunkt von IaaS und beim Einsatz in Integrationsarchitekturen. Die Ausführungsumgebung muss als Plattform realisiert sein, das heißt, sie muss über lange Zeiträume ununterbrochen laufen und benötigt vielfältige Schnittstellen. Diese müssen einen entfernten Zugriff zulassen und sollen den Anwender bei der Installation und der Steuerung von Konvertierungen sowie bei deren Überwachung unterstützen.

Der Universal Data Converter im Überblick

Nachdem damit die Grundlagen bekannt sind, soll nun die Praxis von Datenintegrationsprojekten mit dem Universal Data Converter veranschaulicht werden. Dabei kann nicht auf sämtliche Funktionsmerkmale eingegangen werden. Eine Liste der Möglichkeiten des Universal Data Converter finden Sie im Internet [1]. Eine Zusammenstellung der unterstützten Basisdatenformate, der bereitgestellten Module sowie der DCDL-Unterstützung finden Sie in den Textkästen „Basisdatenformate“, „Module“ und „DCDL“.

Die für den Anwender wichtigsten sichtbaren Teile des Universal Data Converter fasst Tabelle 1 zusammen.

Der UDC-Designer als Cockpit

Wie schon beschrieben, ist der Designer das Cockpit für die Entwicklung von Konvertierungen, die technische Spezifikation der einzelnen Schritte einer Konvertierung sowie das erste Ausprobieren, Testen und Verpacken der Ergebnisse. Zusammengefasst finden Sie hier alles, was benötigt wird, um ein Projekt von den ersten Schritten bis zum auslieferbaren Zustand ▶

Technische Details (Bild 4)

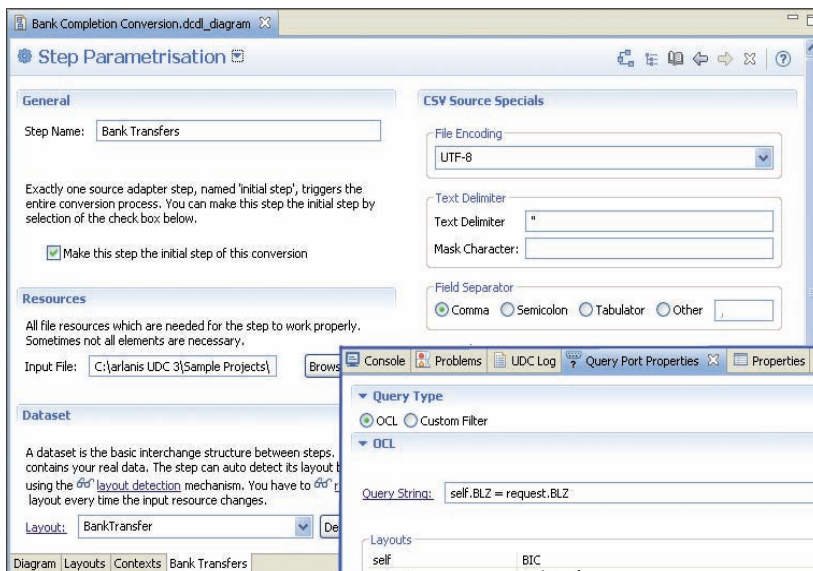


Tabelle 2: Funktionsbereiche des UDC-Designers.

Komponente	Beschreibung
Projekte	Alle UDC-Projekte werden hier übersichtlich verwaltet. Es besteht keine Limitierung der Projekte.
Zeichen- und Arbeitsfläche	In dieser Fläche werden die Konvertierungen und grafischen Daten-Mappings erstellt. Auch werden hier die technischen Details einer Komponente auf der Zeichenfläche spezifiziert. Sie stellen sich einfach ein weißes Blatt Papier sowie eine Menge Buntstifte vor und malen die Konvertierung.
Werkzeuge, Adapter und Module	Vorgefertigte Komponenten und Werkzeuge sind hier zu finden. Jede neue UDC-Version bringt ein paar mehr davon mit.
Übersicht	Hier behalten Sie Übersicht über die Zeichenfläche, auch wenn dort nur ein Ausschnitt zu sehen ist.
Konsole, Logging und Informationen	Alle Informationen während der Laufzeit, die Ergebnisse der Validierung während der Designzeit oder auch die kleinsten Eigenschaften sind hier zu finden.

zu bringen. Der Designer selbst basiert auf einem OSGi/Eclipse-Kern und präsentiert sich nach dem Start in einem aufgeräumten und übersichtlichen Zustand. In Bild 3 sind die einzelnen Bereiche ersichtlich.

Je nachdem welche Aufgabe Sie gerade meistern möchten, bewegen Sie sich in einen der in Tabelle 2 zusammengefassten fünf Bereiche.

Praxisbeispiel: Bank Completion

Der erste Schritt in eine neue Konvertierung ist ein fachlicher Überblick, was einfach zu erreichen ist. Im ersten Projekt sollen eine Menge von Banküberweisungen vervollständigt und dann komplett in einer CSV-Datei (Comma Separated Values) abgelegt werden, und zwar zeilenweise [2], Zeile für Zeile. Das bedeutet, Sie müssen einen Datensatz vervollständigen und diesen dann in eine neue Datei schreiben. Den Ausgang stellt eine CSV-Datei mit Banküberweisungen dar. Diese haben Sie sicherlich schon einmal gesehen und sich eventuell gefragt, was passiert, wenn die Bankleitzahl nicht stimmt. Nun, keine Sorge, wenn Ihre Konvertierung fertig ist (obwohl der Begriff Konvertierung vielleicht nicht ganz richtig ist), wird die Überweisung automatisch geprüft. Als Erstes betrachten Sie die beteiligten Komponenten und tragen diese im Designer in die Zeichenfläche ein. Jetzt haben Sie eine CSV-Datei, mit den Überweisungen, die

Ausgabedatei, eine CSV-Datei in welcher alle gültigen Bankleitzahlen und der Name der dazugehörigen Bank enthalten sind, und eine Java-Transformationskomponente. Im Anschluss sagen Sie, wie welche Komponenten verbunden sind, und schon ist die Konvertierung im Design fertig.

Damit auch die Dokumentation nicht zu kurz kommt, fügen Sie an der einen oder anderen Stelle Bilder und Texte ein. Damit haben Sie auch gleich etwas für Ihre PowerPoint-Präsentation zur Hand. Aber zurück zur Konvertierung: Sie ergänzen jetzt jede Komponente um ihre technischen Details. Das geschieht ganz einfach durch Doppelklick auf die jeweilige Komponente. Welche Daten dort eingegeben werden, ist abhängig vom gewählten Modul. Bild 4 zeigt exemplarisch die Einstellungen für die CSV-Eingabedatei. Sie werden sich natürlich fragen, wie die Datensätze (Überweisungen und Bankliste) miteinander verknüpft werden. Ganz einfach, Sie gehen auf den Query-Port und tragen mittels der Object Constraint Language (OCL) die Bedingung ein (Bild 4). Diese Abfragesprache ermöglicht es Ihnen, völlig neutral vom gewählten Datenformat auch beliebige Verknüpfungen von Datensätzen zu realisieren.

Besonders hervorzuheben ist in dieser Konvertierung die Java-Komponente. In dieser können Sie per Java vollständig die Umsetzung von Eingangs- auf Ausgangsdatsatz steuern. Ihrer Fantasie sind dabei keine Grenzen gesetzt und, da als Editor eine vollständige Java-Entwicklungsumgebung vorhanden ist, auch keine Hindernisse. Nachdem auch die technischen Details hinzugefügt sind, steht der Ausführung nichts mehr im Wege.

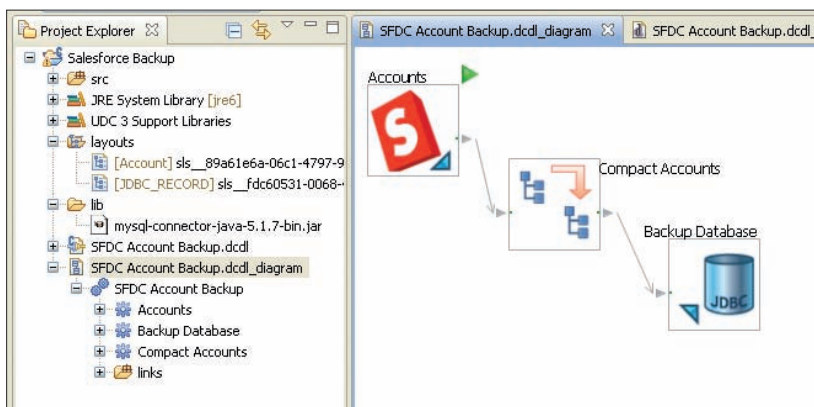
Praxisbeispiel: Salesforce.com-Backup

Im zweiten Projekt müssen Informationen aus einem Salesforce.com-Account-Objekt in einer Datenbank abgelegt werden [4]. Damit es einfacher wird, ist es hier kein Echtzeitprojekt. Sie müssen sich auch nicht darum kümmern, wie

Testversion

Informationen über die verfügbaren Editionen sowie eine Testversion sind über die Internetpräsenz der Arlanis Software AG beziehbar. Vor dem Herunterladen einer Testversion über die Internetseite <http://arlanis.de/de/software/losungen/download.html> müssen Sie sich namentlich mit Firmen- und Kontaktadressen beim Hersteller registrieren.

Überblick über das Integrationsprojekt (Bild 5)



Daten zurück nach Salesforce.com übernommen werden. Auch in diesem Projekt fangen Sie am besten mit der Übersicht an: Salesforce.com als Quelle auf der Eingangsseite, eine relationales Datenbanksystem als Ziel auf der Ausgangsseite und eine grafische Transformation, um die benötigten Verknüpfungen herzustellen. Den Überblick finden Sie in **Bild 5**.

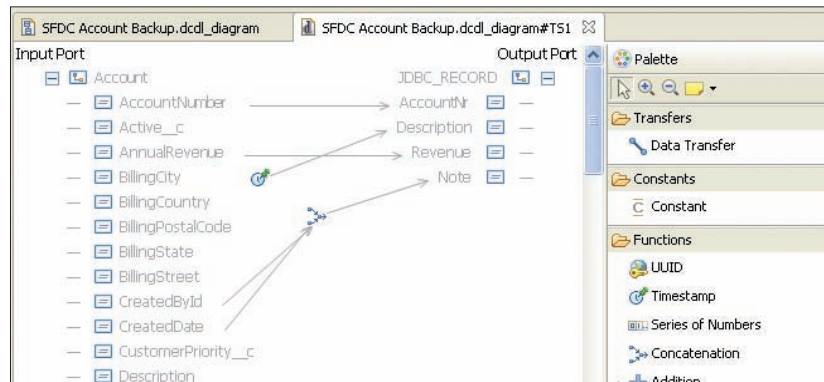
Alternative: Grafisches Daten-Mapping

Der Aufbau der Konvertierung geschieht, wie schon gezeigt, im Designer. Nur dass diesmal eine Salesforce.com-Quelle sowie ein JDBC-Ziel genutzt und konfiguriert werden. Im vorherigen Projekt haben Sie die umfangreichen Skript-Fähigkeiten genutzt oder besser gesagt einen winzigen Teil davon. In diesem Projekt wird eine zweite Form der Transformationskomponente verwendet: das grafische Daten-Mapping. Im Prinzip macht das grafische Mapping das Gleiche wie eine Skript-Komponente, nur eben grafisch. Interessant und spannend ist es, die Grenzlinie zu finden, wann eine der beiden Komponenten eingesetzt wird. So kann die gezeigte Aufgabe leicht grafisch erfüllt werden. Aber denken Sie mal an eine Edifact-nach-XML-Transformation (n*100 Felder auf der Eingangsseite, n*100 Felder auf der Ausgangsseite) und vergessen Sie nicht, die Größe Ihres Monitor in die Betrachtung einfließen zu lassen. Nun, ein Doppelklick auf die Transformationskomponente öffnet das grafische Daten-Mapping, wie in **Bild 6** zu sehen ist.

UDC-Plattform als Management Web Service

Damit haben Sie gesehen, wie eine Konvertierung entworfen, getestet und auch ausgeführt wird. So weit, so gut. Jedoch wollen Sie in vielen Fällen nicht erst den Designer öffnen und dann per Hand die Ausführung starten. Im Zeitalter des Cloud Computing bietet der UDC 3.0 eine Plattform, welche beim Anwender installierbar ist. Alternativ kann sich ein Anwender auch eine bereits vorhandene UDC-Plattform-Installation suchen und sich dort etwas Platz mieten. Aber zurück zum Designer: Aus dem Designer werden die einzelnen Konvertierungen als Deployable Conversion Archives (verteilbare Konvertierungsarchive) exportiert und im Anschluss in die Plattform installiert. Dort können diese dann überwacht, gestartet oder zeitplanmäßig ausgeführt werden, wie es in **Bild 7** zu sehen ist.

Die Plattform selbst stellt einen Verwaltungs-Webservice (Management Web Service) bereit. Somit ist die Integration in Anwendungen und auch in die Webseiten des Anwenders problem-



Grafisches Daten-Mapping (**Bild 6**)

los möglich. Gleichzeitig wird jede einzelne Konvertierung ebenfalls als Webservice bereitgestellt und kann somit in beliebigen Szenarien betrieben oder auch in Geschäftsprozesse eingebettet werden.

Fazit: Komfortabel und leistungsfähig

Der Universal Data Converter stellt durch seine Architektur und die einzelnen Teile eine komfortable und leistungsfähige Lösung für die Datenintegration bereit, die selbstverständlich auch für die Durchführung von Migrationsprojekten geeignet ist. Dabei skaliert das System von einfachen einmaligen Konvertierungen bis hin zum Bestandteil des Cloud Computing. Die UDC-3.0-Plattform erfüllt alle Voraussetzungen, um in einer modernen, vernetzten Welt die Rolle des Integration-as-a-Service-Parts zu übernehmen. Auf den Webseiten zum UDC sind verschiedene weitere Informationen zu finden. [am]

- [1] Arlanis Software AG – www.arlanis.de
- [2] Tutorial „Bank Completion“ aus der UDC-Dokumentation
- [3] Tutorial „Salesforce.com Backup“ aus der UDC-Dokumentation
- [4] Salesforce.com; www.salesforce.com/de

Die UDC-V3-Plattform (**Bild 7**)

