

**Benutzerfreundliche Aufbereitung und Validierung von XML-Daten mit JAXFront.**

## **Intelligente XML-Clients**

**Anwender sind es gewohnt für die Datenerfassung komfortable grafische Oberflächen zur Verfügung zu haben. Immer häufiger handelt es sich dabei um Daten, die am Ende in einem XML-Format vorliegen sollten. Anstatt nun diese Oberflächen auszuprogrammieren, verfolgt JAXFront den Ansatz der Generierung, das heisst die Benutzeroberflächen werden anhand ihrer Metadaten (XML-Schema) zur Laufzeit generiert und nicht wie sonst üblich programmiert. So kann für jedes XML-Dokument die dazugehörige Oberfläche zur Erfassung der Daten ohne Programmieraufwand dynamisch erzeugt werden.**

XML speichert Daten in einem applikationsunabhängigen Format. Diese fundamentale Eigenschaft ermöglicht es, dass das XML-Format als Vermittler zwischen verschiedenen Applikationen agieren kann. Web Services sind ein aktuelles Beispiel für diese Vermittlerrolle. XML bildet dabei eine universelle Sprache mit deren Hilfe sich unterschiedliche Applikationen untereinander verständigen können. Dies vereinfacht die Integration verschiedener heterogener Applikationen zu einem Gesamtsystem (EAI). Es verwundert deshalb nicht, dass es eine stetig wachsende Zahl von Applikationen gibt, die ihre Daten mit XML beschreiben und so untereinander austauschen.

XML wurde dafür ausgelegt, dass es für Maschinen und Entwickler leicht lesbar ist. Herkömmlichen Anwendern ist es nicht zuzumuten, XML-Formate direkt zu lesen oder zu editieren. Nehmen wir z.B. den Fall an, dass ein Anwender einen neuen Kunden erfassen muss. Damit der erfasste Kunde im XML-Format an ein nachgelagertes System gesendet werden kann, muss eine Lösung programmiert werden, welches das Erfassen von Daten ermöglicht und die Umwandlung in XML übernimmt.

JAXFront schließt die Lücke zwischen Benutzer und XML-Anbindung an ein System, indem es die Generierung der Oberflächen und das Umwandeln der Daten in XML auf der Benutzerseite übernimmt. Dies vereinfacht eine durchgehende, einheitliche Verwendung von XML-Daten vom Server bis zum Client. JAXFront benötigt lediglich die Beschreibung der erwarteten XML-Daten in Form eines XML-Schemas. Allein aufgrund dieser Information generiert JAXFront zur Laufzeit einen „smarten“ Java Client, der aus den Benutzereingaben ein valides XML-Dokument erzeugt. Folgende Abschnitte sollen darüber Aufschluss geben, wie mit Hilfe von JAXFront solche Clients mit minimalem Aufwand erstellt werden können.

## Integration

Da JAXFront für die Erstellung eines Clients ein XML-Schema benötigt, genügt es, wenn der Server das entsprechende Schema dem Client übermittelt. Der generierte JAXFront Client kümmert sich um die Erfassung der Eingaben, das Binding der Daten und das Erstellen eines XML-Dokuments, das den Schema-Anforderungen genügt. Listing 1 illustriert wie wenig Java Code dazu notwendig ist. Zuerst wird das Schema dem JAXFront-Parser übergeben. Dieser setzt auf einem Schema-Parser auf und liefert ein sogenanntes JAXFront Document Object Model (DOM) zurück. Übergibt man dieses DOM einem JAXFront „EditorPanel“ wird eine graphische Oberfläche erzeugt, die es erlaubt, die im Schema beschriebenen XML-Dokumente zu erfassen. Wird beim Erzeugen des DOMs eine XML-Instanz mitgegeben, dann werden diese angezeigt und können sofort über die Oberfläche editiert werden. Beim „EditorPanel“ handelt es sich um ein Java Swing JPanel, welches in eine eigene Applikation eingebaut oder als eigenständige Applikation in einem Fenster verwendet werden kann.

```
URL xsdURL = new URL("file:///c:\\temp\\mySchema.xsd");
Document dom = DOMBuilder.getInstance().build(xsdURL);
JFrame frame = new JFrame("JAXFront - Intelligente XML Clients");
EditorPanel editorPanel = new EditorPanel(dom.getRootType(),frame);

frame.getContentPane().setLayout(new java.awt.BorderLayout());
frame.getContentPane().add(editorPanel, java.awt.BorderLayout.CENTER);
frame.setSize(640, 480);
frame.setVisible(true);
```

Listing 1

Man kann JAXFront auch dazu nutzen, nur Teile eines Schemas zu visualisieren. Dies erlaubt es, die Erfassung der Daten flexibel in eine bestehende Java-Applikation einzubinden. So können z.B. auch eigene Layout Manager verwendet werden, um die einzelnen Komponenten anzuordnen. Im Code Beispiel (Listing 2) sieht man, dass die persönlichen Angaben und die Liste der Kinder getrennt visualisiert werden. Dazu gibt man lediglich den Knoten des Schemas an, der visualisiert werden soll. JAXFront liefert eine JComponent zurück, welche in die eigene Applikation eingebettet werden kann. Es handelt sich dabei um „intelligente“ Komponenten, die das Binding und die Validierung übernehmen, so dass der Entwickler sich nur noch um die visuelle Einbettung kümmern muss. Die vom Anwender erfassten Daten im XML-Format erhält man, indem man auf dem JAXFront-DOM die Funktion „serialize“ aufruft.

```
JPanel myOwnPanel = new JPanel();
myOwnPanel.setLayout(new BorderLayout(myOwnPanel, BorderLayout.Y_AXIS));
JComponent parentComponent =
(JComponent)TypeVisualizerFactory.getInstance().getVisualizer(dom.getRootType().
getChild("Parent"));
JComponent childrenComponent =
(JComponent)TypeVisualizerFactory.getInstance().getVisualizer(dom.getRootType().
getChild("Children"));
JComponent paymentComponent =
(JComponent)TypeVisualizerFactory.getInstance().getVisualizer(dom.getRootType().
getChild("PaymentMethod"));
myOwnPanel.add(parentComponent);
myOwnPanel.add(childrenComponent);
JSplitPane splitPane = new JSplitPane(JSplitPane.VERTICAL_SPLIT, myOwnPanel,
paymentComponent);

frame.getContentPane().add(splitPane, java.awt.BorderLayout.CENTER);
frame.setSize(640, 480);
frame.setVisible(true);
```

*Listing 2*

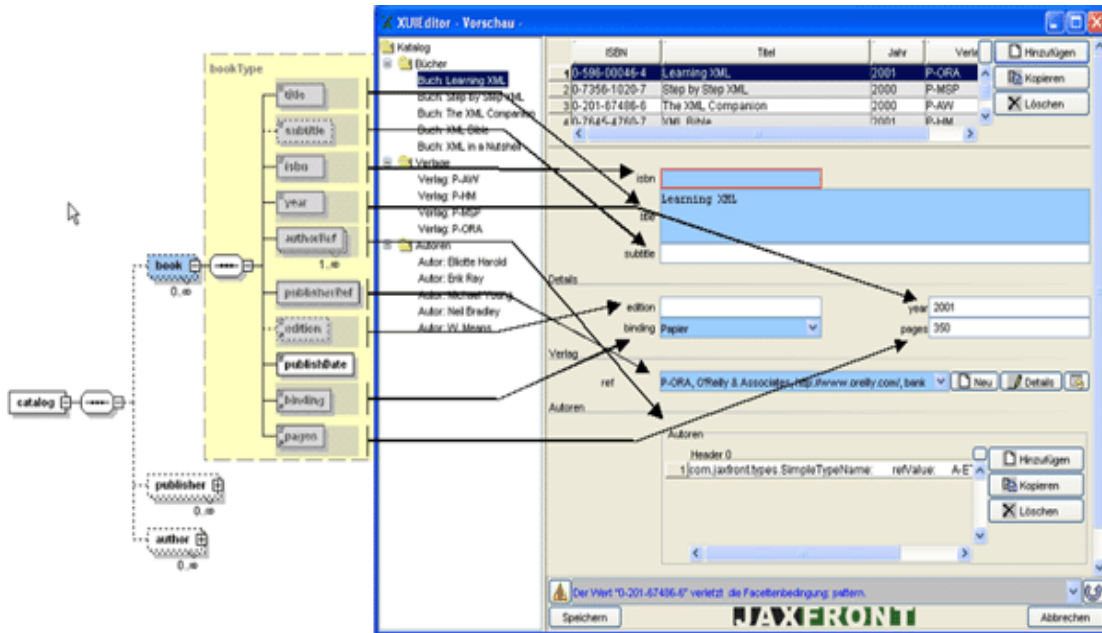
Damit der Client einsatzfähig wird, muss man nur noch festlegen, wie das Schema zum Client übermittelt werden soll und in welche Form der Client das erzeugte XML zum Server zurückschickt. Hier ist der Entwickler frei, ein Kommunikations-Verfahren zu wählen, welches sich am besten in das bestehende System einbetten lässt.

## **Flexibilität**

Da der Client die Oberfläche zur Laufzeit aufgrund der Schema-Informationen erzeugt, lassen sich mit dem gleichen Client beliebige Oberflächen erzeugen. Möchte man ein bestehendes Produkt anpassen, lassen sich die Änderungen bequem im Schema vornehmen. Benötigt man z.B. auch die Adresse eines Kunden, wird diese im Schema eingefügt. Wenn der JAXFront-Client das Schema neu lädt, werden die Änderungen automatisch in der Oberfläche nachgezogen. Dies minimiert den Aufwand bei Änderungen und ermöglicht ein „rapid prototyping“.

## **Anpassen**

Möchte man das „Default Rendering“ der JAXFront-Oberflächen für das entsprechende Produkt optimieren, lassen sich Anpassungen vornehmen. Diese können in einem separaten Editor vorgenommen werden. Da die Änderungen auf dem „Default Rendering“ basieren, muss nicht das ganze Layout der Oberfläche neu gestaltet werden, sondern man ändert lediglich die Einstellungen, welche von Bedeutung sind. Im XUI-Editor (eXtensible User Interface) kann man die Anordnung der einzelnen Eingabe-Komponenten ändern und nach eigenen Bedürfnissen gestalten. Die Komponenten lassen sich individuell beschriften (auch mehrsprachig). Neben den grafischen Einstellungen lassen sich auch Regeln definieren, die eine Fülle von Aktionen auslösen können.



Verknüpfung von XML Schema Model mit Oberfläche

So können Elemente der Oberfläche zur Laufzeit ein- und ausgeblendet, Felder berechnet oder Hilfetexte angezeigt werden. Diese Anpassungen werden in einem separaten XML-Dokument mit der Endung „xui“ abgelegt. Zu einem Schema können verschiedene Oberflächen existieren, die in eigenen „xui“-Files abgespeichert werden. Übergibt man dieses File zusammen mit einem Schema an die JAXFRONT Rendering Engine, dann wird eine optimierte Oberfläche erzeugt. Dieser Mechanismus erlaubt es für spezifische Benutzergruppen angepasste Oberflächen zur Verfügung zu stellen.

## Zusammenfassung

In Firmen, in denen heute XML und XML-Schemas eingesetzt werden, vereinfacht JAXFRONT das Erstellen von Frontends zur Bearbeitung der XML-Daten. Die erzeugten Oberflächen können sehr einfach in bestehende Java Anwendungen eingebunden werden oder, falls gewünscht, in HTML oder PDF dargestellt werden.

Die Stärke von JAXFRONT besteht darin, dass es den Aufwand zur Pflege und Erstellung von Clients minimiert. JAXFRONT bietet hierfür eine flexible Lösung mit klaren Schnittstellen (XML-Schema), welche den Entwicklungsprozess schlanker macht und es dem Entwickler erlaubt, sich stärker auf die Geschäftslogik und der damit verbundenen Prozesse zu fokussieren.