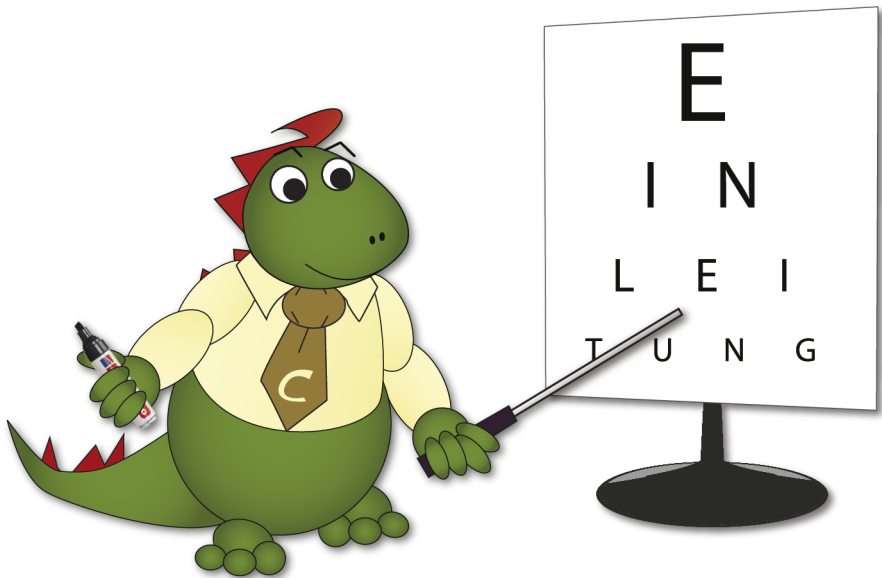


Vorwort



Vorwort

Herzlich willkommen zum ersten Buch einer zweiteiligen Reihe zur Transformation von XML mit Hilfe von XSLT und XSL-FO, auch bekannt unter der zusammenfassenden Abkürzung XSL (eXtensible Stylesheet Language (for Transformations)). In diesem ersten Band erfahren Sie alles Wesentliche, um mit Hilfe von XSLT 1.0 und 2.0 Algorithmen zu entwerfen, mit deren Hilfe XML-Dateien in HTML, Text und andere XML-Formate übertragen werden können. Darüber hinaus lernen Sie XPath 1.0 und 2.0 kennen, um in XML-Dokumenten zu navigieren, Knoten zu lokalisieren und Funktionen aus der XPath/XQuery-Bibliothek anzuwenden. Das Buch schließt mit einer Darstellung zu XQuery, einer Alternative zu XSLT, welche den Aspekt der Abfrage und nicht den der Transformation von XML-Daten in den Vordergrund stellt, ab.

Konzeption der Reihe

Die Aufteilung in Bücher und Kapitel lässt sich so begründen, dass nicht jeder Leser beide Werke benötigt und er vielleicht eine Auswahl treffen möchte. Einige möchten möglicherweise auch nach und nach beide Bände kaufen und nicht von vorneherein ein erschreckend umfangreiches Buch in der Hand halten.

- Der erste Band stellt die Syntax von XSLT dar. Es umfasst die Standards von XSLT 1.0 mit dem Titel XSL Transformations (XSLT), Version 1.0, W3C Recommendation 16 November 1999 unter <http://www.w3.org/TR/xslt> und XPath 1.0 mit dem Titel XML Path Language (XPath), Version 1.0, W3C Recommendation 16 November 1999 unter <http://www.w3.org/TR/xpath> aus Version 1 und die Standards XSLT 2.0 mit dem Titel XSL Transformations (XSLT), Version 2.0, W3C Working Draft 12 November 2003 unter <http://www.w3.org/TR/xslt20/> und XPath 2.0 mit dem Titel XML Path Language (XPath) 2.0, W3C Working Draft 23 July 2004 unter <http://www.w3.org/TR/xpath20/>. Sie lernen in diesem Band daher alle Techniken kennen, welche zur Erzeugung von HTML, Text und XML und zur Abfrage von XML-Dokumenten mit XPath notwendig ist. Die Grundlagen werden im ersten Teil geschaffen. Der zweite Teil dieses Bandes stellt neben fortgeschrittenen Techniken eine Reihe von Einsatzbeispielen vor. Er kombiniert die einzelnen vorgestellten Standards mit allgemeinen Techniken, die nur die Transfor-

mation betreffen, mit Ideen, wie XSLT-Anwendungen aufgebaut werden können, und solchen Techniken wie die Verwendung einer Datenbank, um XML- und sogar XSLT-Daten abzuspeichern und dynamisch für Transformationsabläufe zusammensetzen.

- Der zweite Band stellt die Syntax von XSL-FO 1.0 bzw. XSL 1.0 mit dem Titel Extensible Stylesheet Language (XSL), Version 1.0, W3C Recommendation 15 October 2001 unter <http://www.w3.org/TR/xsl/> dar. Sie lernen in diesem Band, wie Druckerzeugnisse wie z.B. PDF-Dateien aus XML-Daten erzeugt werden. Dies erfordert einen eigenen Standard mit einer Vielzahl an unterschiedlichen Elementen, um Seitenbereiche, Seitenverläufe, Text- und Absatzformate sowie zusätzliche Dokumenteigenschaften anzugeben wie Inhaltsverzeichnisse.
- Weitere Bände zur XML-Verwendung in Software sind ebenfalls bei Comelio Medien erhältlich. Sie betreffen neben XSLT auch die Umwandlung und Erstellung von XML in Programmiersprachen sowie die Verwendung von XML in Datenbanken.

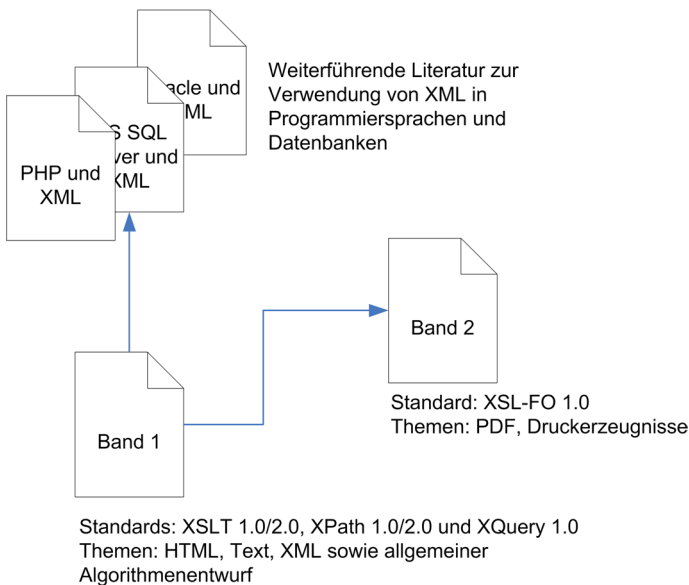


Abbildung v.1: Aufbau der Reihe

Leserkreis

Die zwei Bücher richten sich an Leser, die in jedem Fall bereits grundlegende Kenntnisse über XML und dessen Einsatzbereiche haben. Grundlegende Kenntnisse sollten Antworten auf die folgenden Fragen umfassen:

- Was haben HTML und XML gemeinsam? – Sie stellen beide eine Syntax dar, in der Inhalte über Bezeichner in spitzen Klammern ausgezeichnet werden. Diese Bezeichner können jeweils so genannte Attribute enthalten, die ebenfalls Textwerte nach einem Gleichheitszeichen und innerhalb von Anführungszeichen aufweisen können. Es entstehen verschachtelte Strukturen, die in einem Baum wiedergegeben werden können. Regelungen zur Benennung folgen dem Standard Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04 February 2004 unter <http://www.w3c.org/TR/2004/REC-xml-20040204/>. HTML kann so verwendet werden, dass kein wohlgeformtes XML entsteht.
- Ist HTML auch XML? – HTML kann so verwendet werden, dass wohlgeformtes XML entsteht. XHTML überführt den HTML-Standard in einen XML-Standard durch einige weitere Syntaxregeln, die einzuhalten sind, damit wohlgeformte XML-Dokumente entstehen. In diesem Sinne stellt XHTML eine spezielle Grammatik dar, mit deren Hilfe Webseiten ausgezeichnet werden können, während andere XML-Formate andere Strukturen abbilden, darunter auch eigene, die nicht von einer fremden Organisation entwickelt wurden.
- Was ist die Document Type Definition? – Eine DTD ist ein Regeldokument bzw. eine Ansammlung von Strukturvorgaben/Regeln, deren Syntax im Standard Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04 February 2004 unter <http://www.w3c.org/TR/2004/REC-xml-20040204/> beschrieben werden. Mit ihrer Hilfe lässt sich angeben, wie die Elemente und Attribute eines XML-Instanzdokumentes heißen, wie sie verschachtelt sind und in welcher Hierarchie und Häufigkeit sie auftreten. Die DTD besitzt selbst keine XML-Struktur.
- Was ist XML Schema? – Ein XML-Schema-Dokument ist ein Regeldokument bzw. eine Ansammlung von Strukturvorgaben/Regeln, deren Syntax im XML Schema Part 1: Structures, W3C Recommendation 2 May 2001 unter <http://www.w3c.org/TR/xmlschema-1/> beschrieben werden. Diese Syntax stellt selbst auch wiederum ein XML-Format dar. Sie weist einige Vorteile gegenüber der DTD auf. Zum Beispiel ist es möglich, Datentypen (XML Schema Part 2: Datatypes, W3C Recommendation 02 May 2001 unter <http://www.w3c.org/TR/xmlschema-2/>) für Elemente anzugeben oder objekt-

orientierte Strukturen für die Angabe von Regelungen zu verwenden. Mit ihrer Hilfe lässt sich angeben, wie die Elemente und Attribute eines XML-Instanzdokumentes heißen, wie sie verschachtelt sind, welche Datentypen ihre Inhalte haben und in welcher Hierarchie und Häufigkeit sie auftreten.

Autoren

Ein solches Werk schreibt man nicht alleine, sondern teilt sich die Arbeit nach Interessen und Schwerpunkten auf. Die Autoren arbeiten in verschiedenen Positionen bei der Comelio GmbH (www.comelio.com) und haben bereits gemeinsam oder alleine verschiedene Bücher zu ihren Werkzeugen veröffentlicht.

- Marco Skulschus arbeitet als Projektleiter und Dozent und beschäftigt sich mit Berichts- und Expertensystemen auf Basis von MS SQL Server und Oracle. Er interessiert sich besonders für Ontologien und Data Mining-Techniken.
- Marcus Wiederstein arbeitet als Projektleiter für Projekte mit Microsoft-Produkten wie MS BizTalk Server, MS Sharepoint Portal Server oder MS Project Server sowie den Business Intelligence-Produkten.
- Sarah Winterstone arbeitet als Entwicklerin bei der Comelio, Inc. in Miami, FL und programmiert kaufmännische Anwendungen mit .NET und setzt dabei vielfältig XML-Technologien ein. Ihr Spezialgebiet ist die Integration von XML in Datenbanken bzw. die Entwicklung von Import-/Export-Schnittstellen auf XML-Basis.

Aufbau des Buchs

Dieses Buch soll Sie durch die vollständige Syntax von XSLT führen und dabei auch einige Anregungen für die konkrete Anwendungsentwicklung mit dieser interessanten und flexiblen Technologie geben.

1. Das erste Kapitel stellt die Funktionsweise und den Einsatzbereich von XSLT dar. Es geht darauf ein, welche verschiedenen Alternativen zu XSLT existieren und zeigt, in welchen Fällen XSLT die bessere Wahl ist.
2. Das zweite Kapitel führt in die verschiedenen Vorlagenarten von XSLT ein. Da XSLT eine vorlagengetriebene Sprache ist, in denen die einzelnen Umwandlungsregeln

in `xsl:template`-Elementen angegeben werden, kann man hier vier verschiedene Arten unterscheiden, die der Reihe nach mit Beispielen vorgestellt werden. Dies umfasst die Vorlagen-Regeln, wobei passende Umwandlungsangaben anhand eines XPath-Ausdrucks ausgewählt werden, die benannten Vorlagen, welche direkt über ihren Namen aufgerufen werden können und mit denen man wiederkehrende Routinen erstellen kann, Attributwervorlagen für die Erzeugung von Attributwerten im Ausgabestrom und schließlich Vorlagenmodi, mit denen man für das gleiche XPath-Muster verschiedene Vorlagen-Regeln erstellen kann.

3. Das dritte Kapitel führt dann umfassend in die XPath-Syntax ein. Mit Hilfe dieser so genannten Pfadbeschreibungssprache ist es überhaupt erst möglich, Knoten zu filtern und im Dokument zu lokalisieren. Man unterscheidet hierbei relative und absolute Ausdrücke, die in Fallunterscheidungen, bei der Formulierung von Vorlagen-Regeln sowie der Zuweisung von Werten für Parametern und Variablen zum Einsatz kommen. XPath besitzt eine Basis-Syntax, die entlang von Achsen im XML-Eingabestrom Knoten lokalisieren kann, und eine Funktionsbibliothek, die in diesem Kapitel für Version 1.0 vorgestellt wird.
4. Das vierte Kapitel stellt die Möglichkeiten dar, in XSLT Kontrollanweisungen für Fallunterscheidungen und Schleifen zu verwenden. Das Kapitel beginnt mit einer Darstellung, wie XPath für die Knotenauswahl und Filterung verwendet werden kann. Danach zeigt es die beiden Elemente für Kontrollstrukturen `xsl:if` und `xsl:choose` sowie die Anweisung `xsl:for-each` für Wiederholungen.
5. Das fünfte Kapitel stellt die Elemente für Sortierungen, Nummerierungen und Gruppierungen in den Vordergrund. Mit dem `xsl:sort`-Element kann man einstufig und mehrstufig in einer ermittelten Knotenmenge sortieren und die einzelnen Knoten in dieser Sortierung verarbeiten. Für die Nummerierung kommt das `xsl:number`-Element zum Einsatz, in dem nahezu beliebige Optionen für das Zahlenformat bereitstehen. Gruppierungen erklärt das Kapitel schließlich sowohl für die Version 1.0 mit Hilfe der hier gängigen Methoden und für die Version 2.0 mit dem `xsl:for-each-group`-Element, dessen Einstellungen die Möglichkeiten, die die 1.0-Techniken bieten, noch weit übersteigen.
6. Das sechste Kapitel zeigt, wie man in XSLT Variablen und Parameter verwenden kann, um Werte im Rahmen einer Transformation zwischenspeichern oder Werte an Vorlagen zu übergeben. Mit dem Element `xsl:variable` lassen sich globale und temporäre Werte speichern. Das Element `xsl:param` hingegen erlaubt, bei der lokalen Verwendung Vorlagen zu parametrisieren, während ein globaler Einsatz einen so ge-

nannten Stylesheet-Parameter festlegt, mit dem man Werte aus der aufrufenden Umgebung an die Transformation übergeben kann.

7. Das siebte Kapitel behandelt eine Reihe von speziellen Ausgabeanweisungen, mit denen für unterschiedliche Formatierungen und spezielle Einsatzbereiche Vorgaben für den Ausgabestrom getroffen werden können. Manchmal stellen diese Techniken Vereinfachungen von aus Basisfunktionalitäten zusammengesetzten Kombinationen dar. Man lernt hier, wie man mehrere XML-Dateien gleichzeitig verarbeiten und erzeugen kann, oder wie Formatierungsanweisungen global vorgegeben werden können.
8. Das achte Kapitel verlässt den Bereich der HTML-Erzeugung und konzentriert sich dagegen auf die Besonderheiten, die bei der Generierung von XML und Text zu beachten sind. In Beispielen wird hier gezeigt, wie man aus XML Schema dynamisch HTML-Formulare erstellt, XML-Strukturen ändert oder aus XML Text-Dateien wie bspw. CSV erzeugt. Dabei setzt man eine Reihe von anderen Techniken wie Variablen und Parametern aus früheren Kapiteln wieder ein.
9. Das neunte Kapitel erklärt die Elemente, mit denen Stylesheets in Module zerlegt und dann wieder zusammen gesetzt werden können. Hier sind in XSLT zwei Elemente mit feinen Unterschieden verfügbar. Für den so genannten Import verwendet man `xsl:import`, für die Einbindung `xsl:include`. Sie unterscheiden sich in ihrem Verhalten, wie sie lokale Vorlagen überschreiben.
10. Das zehnte Kapitel zeigt in verschiedenen Beispielen, wie man effektiv Fehler suchen und Nachrichten während der Transformation ausgeben kann.
11. Das elfte Kapitel diskutiert Möglichkeiten und Bedingungen, XML-Dateien anhand ihrer Struktur und nicht anhand der Namen von Elementen und Attributen zu verarbeiten. Dies schärft den Blick, besonders dynamische und wiederverwendbare Algorithmen in XSLT zu entwickeln.
12. Das zwölfte Kapitel ergänzt die verschiedenen Vorlagentechniken aus XSLT 1.0 um eine Neuerung in XSLT 2.0, die es erlaubt, Stylesheet-Funktionen zu erstellen. Sie können in XPath-Ausdrücken für die Durchführung von Berechnungen oder Ausgaben zum Einsatz kommen und verkürzen die Snytax im Vergleich zu benannten Vorlagen erheblich. Diese Kapitel stellt die 1.0-Vorgehensweise, die benannte Vorlagen einsetzte, mit der 2.0-Technik über `xsl:function` gegenüber.

13. Das dreizehnte Kapitel bindet zusätzlich die XML Schema-Datei ein, welche die umzuwandelnde XML-Datei beschreibt. Durch diese neue 2.0-Technik ist es möglich, eigene in XML Schema erstellte Datentypen zu testen oder abzufragen und dabei eine Reihe von XPath 2.0-Operatoren zu verwenden. So kann man Umwandlungen einrichten, die struktur- und datentypgetrieben sind.
14. Das vierzehnte Kapitel arbeitet die Funktionsbibliothek von XPath 2.0 mit thematisch sortierten Beispielen durch, um so die vielen neuen nützlichen Funktionen und Vereinfachungen gegenüber der Vorgängerversion zu zeigen.
15. Das fünfzehnte Kapitel arbeitet gleichermaßen eine Funktionsbibliothek durch, wobei dies in diesem Fall die in XSLT vorhandenen Funktionen sind, um Entitäten aus DTDs oder Textdateien zu verarbeiten oder Schlüssel zu erzeugen und abzufragen.
16. Das sechzehnte Kapitel behandelt XML-Dateien, die Daten enthält, welche wie in eine relationalen Datenbank aufgebaut sind und daher über entsprechende Abfragetechniken in XSLT oder XPath wieder zusammengeführt werden müssen. In einer Reihe von Beispielen lernt man, wie man innere, äußere und selbstbezügliche Verknüpfungen sowie Mengenuntersuchungen auf solchermaßen aus einer DB abgerufenen Strukturen durchführt.
17. Das siebzehnte Kapitel diskutiert fortgeschrittene Techniken und Möglichkeiten, wie man XSLT dynamisch aus XML Schema oder aufgrund von Annahmen erzeugen oder aus mehreren XSLT-Dateien zusammensetzen kann. Als weitere Möglichkeit geht es darauf ein, wie man eine Datenbank einsetzen kann, um XSLT, XML Schema und XML in Spalten zu speichern und dann bei Bedarf zusammensetzen.
18. Das achtzehnte Kapitel zeigt eine neue 2.0-Möglichkeit, in XSLT und XPath reguläre Ausdrücke zu verwenden, um Daten auszuwerten und bedingte Umwandlungen einzurichten.
19. Das neunzehnte Kapitel geht auf eine alternative Syntax ein, mit der XSLT vermieden und abfrageähnliche Transformationen in XQuery, dem SQL für XML, umgesetzt werden können. Das Kapitel umfasst thematisch dabei die Darstellung der Syntax zur Abfrage, die Verwendung von Funktionen, die Formulierung von Variablen und ihre Auswertung sowie die verschiedenen Möglichkeiten, wie man das Ausgabeformat strukturieren kann.

20. Das zwanzigste Kapitel schließlich erklärt, wie man die verschiedenen XSLT-, XPath- und XQuery-Techniken in den Programmiersprachen Java, .NET, PHP sowie in den Datenbanken Oracle mit PL/SQL und in Microsoft SQL Server mit T-SQL nutzen kann. Dies umfasst die Aufgaben, eine Transformation durchzuführen, Parameter an das Stylesheet zu übergeben oder externe Funktionen/Methoden der Sprache in XSLT aufzurufen sowie Abfragen durchzuführen.

Weitere Lern-Ressourcen

Online-Material vom Verlag:

Alle Übungsdateien in XML, XSLT, XSL-FO und entsprechende Ergebnisse finden Sie auf der Verlagsseite www.comelio-medien.com. Um die Ergebnisse von Transformationen in HTML zu testen, brauchen Sie nur einen aktuellen, XSLT-fähigen Internet-Browser, in dem Sie die XML-Dateien öffnen. Alternativ können Sie die sehr weit verbreitete Software von Altova XMLSpy® (Testversion kostenlos erhältlich) von www.altova.com verwenden. Auch das MS Visual Studio erlaubt die Verwendung von XSLT. Alternativ können Sie Ihre Skripte auch in einem beliebigen Text-Editor erstellen und sie lediglich mit einem .NET/Java-XSLT-Prozessor oder der Programmiersprache Ihrer Wahl umwandeln.

Weitere Bücher von Comelio Medien:

- XSL-FO, Marco Skulschus, Jan Kozik, Marcus Wiederstein, ISBN 978-3-939701-17-0
- MS SQL Server 2005 – XML und SOAP-Webservices, Marco Skulschus, Jan Kozik, Alexander Kapitanovskyy, ISBN 978-3-939701-03-3
- PHP und XML, Marco Skulschus, ISBN 978-3-939701-00-2
- Oracle, PL/SQL und XML, Marco Skulschus, ISBN 978-3-939701-10-1

Online-Material vom W3C:

- XSL Transformations (XSLT), Version 1.0, W3C Recommendation 16 November 1999 unter <http://www.w3.org/TR/xslt>
- XML Path Language (XPath), Version 1.0, W3C Recommendation 16 November 1999 unter <http://www.w3.org/TR/xpath>

- XSLT 2.0 mit dem Titel XSL Transformations (XSLT) Version 2.0, W3C Recommendation 23 January 2007 unter <http://www.w3.org/TR/xslt20/>
- XML Path Language (XPath) 2.0, W3C Recommendation 23 January 2007 unter <http://www.w3.org/TR/xpath20/>
- Extensible Stylesheet Language (XSL), Version 1.0, W3C Recommendation 15 October 2001 unter <http://www.w3.org/TR/xsl/>
- Extensible Markup Language (XML) 1.0 (Third Edition), W3C Recommendation 04 February 2004 unter <http://www.w3c.org/TR/2004/REC-xml-20040204/>
- XML Schema Part 1: Structures, W3C Recommendation 02 May 2001 unter <http://www.w3c.org/TR/xmlschema-1/>
- XML Schema Part 2: Datatypes, W3C Recommendation 02 May 2001 unter <http://www.w3c.org/TR/xmlschema-2/>

Textgestaltung

Wie in allen Informatik-Büchern, finden Sie auch in diesem verschiedene typografische Hervorhebungen für *wichtige Begriffe* oder Quelltext sowie Internetadressen oder Dateinamen.

Quelltextblöcke haben besondere Formatierungen.

Gerade im XML-Bereich ist es mit Hilfe geeigneter Editoren besonders einfach, korrekte und formschöne Formatierungen / Einrückungen zu erreichen. Wir möchten Sie an dieser Stelle darauf hinweisen, dass die Formschönheit bei der Verwendung von `xsl:attribute` in Quelltexten eine geringere Bedeutung hatte als die korrekte Verwendung dieses Elements. Damit keine ungewünschten Leerzeichen / Absatzmarken in die Ausgabe gelangen, verzichten wir komplett auf zusätzliche Einrückungen, die eigentlich aufgrund des Zeilenumbruchs notwendig wären. Wenn an diesen Stellen also der Quelltext einmal nicht besonders gut formatiert ist, dann ist dies mit Überlegung geschehen.

Kontakt zu Autor und Verlag

Sie erreichen das Sekretariat der Autoren unter der E-Mail-Adresse info@comelio.com. Von dort werden Ihre Emails dann weitergeleitet. Die Webseite des Verlags finden Sie unter der Adresse <http://www.comelio-medien.com>, während Sie auf der Unternehmenswebseite folgende weitere Informationen finden:

- XML-Artikel: <http://www.comelio-medien.com/comelio-blog>
- XML-Seminare: <http://www.comelio-seminare.com/seminare/titel/xml>

Auch in der realen Welt ist der Verlag zu erreichen. Die Hauptzentrale befindet sich in Berlin, Deutschland.

Comelio GmbH, Goethestr. 34, D-13086 Berlin