

ÜBERSICHT ÜBER OFFICE OPEN XML



ECMA TC45
TOM NGO (NEXTPAGE), EDITOR

1 EINFÜHRUNG

Office Open XML (Open XML) ist ein vorgeschlagener, internationaler offener Standard für Textverarbeitungsdokumente, Präsentationen und Tabellenkalkulationen. Der Standard ermöglicht eine beliebige Implementierung in mehreren Anwendungen auf mehreren Plattformen. Die Veröffentlichung dieses Standards unterstützt Organisationen, die Anwendungen für dieses Format implementieren möchten, gewerbliche und behördliche Einrichtungen, die diese Software beschaffen, und Bildungseinrichtungen oder Autoren, die dieses Format vermitteln. Letztlich profitieren alle Anwender von einem XML-Standard für ihre Dokumente. Zu den Vorzügen zählen beispielsweise die Stabilität, Beständigkeit, Interoperabilität und die Weiterentwicklung dieses Standards.

Die Arbeit an der Standardisierung von Open XML wurde vom Ecma International-Fachausschuss TC45 geleistet. Dieser Gruppe gehören Vertreter von Apple, Barclays Capital, BP, The British Library, Essilor, Intel, Microsoft, NextPage, Novell, Statoil, Toshiba sowie der United States Library of Congress (1) an.

Dieses Whitepaper enthält eine Zusammenfassung zu Open XML. Dieses Dokument vermittelt folgende Kenntnisse:

- Verständnis des Zwecks von Open XML und der Spezifikationsstruktur
- Kenntnisse der Eigenschaften: Umgang mit Abwärtskompatibilität, Beständigkeit, Erweiterbarkeit, individuellen Schemas, Teilmengen, mehreren Plattformen, Internationalisierung und Verfügbarkeit
- Erlernen der übergeordneten Struktur einer Open XML-Datei und schnelles Navigieren durch alle Teile der Spezifikation, aus der Sie weitere Details benötigen

2 ZWECK DES STANDARDS

Open XML wurde von Anfang an so konzipiert, dass alle bereits bestehenden Textverarbeitungsdokumente, Präsentationen und Tabellenkalkulationen, die im binären Format der Microsoft Corporation codiert sind, originalgetreu wiedergegeben werden können. Die Standardisierung umfasste die Abbildung der erforderlichen Korpusdarstellungsfunktionen in XML, deren Erweiterung, die Bereitstellung einer detaillierten Dokumentation und die Unterstützung von Interoperabilität. Während dieses Dokument verfasst wird, erstellen mehr als 400 Millionen Benutzer Dokumente im binären Format (Schätzungen gehen sogar von 40 Milliarden aus). Jedes Jahr kommen weitere Milliarden Dokumente hinzu.

Die ursprünglichen binären Formate dieser Dateien wurden in einer Zeit entwickelt, als Speicherplatz kostbar und die Verarbeitungszeit ausschlaggebend für die Benutzerfreundlichkeit war. Sie basierten auf der direkten Serialisierung der speicherinternen Datenstrukturen von Microsoft® Office®-Anwendungen. Moderne Hardware-, Netzwerk- und Standardinfrastrukturen (vor allem für XML) ermöglichen neue Herangehensweisen, mit denen die Implementierungen durch mehrere Anbieter auf mehreren Plattformen gefördert und Weiterentwicklungen ermöglicht werden.

Parallel zu diesen technologischen Fortschritten haben sich die Märkte diversifiziert und bieten inzwischen eine ganz neue Bandbreite an Anwendungsbereichen, die in der recht einfachen Welt der Dokumentverarbeitungsprogramme anfänglich gar nicht vorgesehen waren. Zu diesen neuen Anwendungsbereichen zählen:

- Automatische Erstellung von Dokumenten auf der Grundlage von Geschäftsdaten
- Extraktion von Geschäftsdaten aus Dokumenten und Einfügen dieser Daten in Geschäftsanwendungen
- Ausführung bestimmter Aufgaben für einen kleinen Teil eines Dokuments bei gleichzeitiger Beibehaltung der Bearbeitbarkeit
- Eingabehilfen für Benutzer mit speziellen Anforderungen, z. B. Sehbehinderte
- Ausführung auf verschiedenen Hardwarekomponenten, z. B. auf mobilen Geräten

Der vielleicht wichtigste Aspekt ist wohl die langfristige Aufbewahrung. Wir haben gelernt, Datenmengen mit exponentieller Zuwachsrate zu erzeugen. Bisher wurden diese Informationen mit digitalen Darstellungen codiert, die so eng mit den verwendeten Programmen verbunden sind, dass sich die Daten nach 10 bis 20 Jahren in der Regel kaum mehr ohne erhebliche Verluste lesen lassen. Die Wahrung der finanziellen und intellektuellen Investitionen in diese Dokumente (neue und alte) hat sich zu einer dringlichen Priorität entwickelt.

Diese vier Faktoren – weitreichende Übernahme des binären Formats, technologische Fortschritte, der Markt, der verschiedenste Anwendungen erforderlich macht, und zunehmende Schwierigkeiten bei der langfristigen Aufbewahrung von Dokumenten – haben zu der Notwendigkeit geführt, ein offenes XML-Format zu definieren und die Milliarden von Dokumenten mit möglichst geringem Verlust zu migrieren. Die Standardisierung dieses offenen XML-Formats und seine langfristige Beibehaltung führen dazu, dass sich jede Organisation vertrauensvoll auf die dauerhafte Stabilität der Spezifikation verlassen und sicher sein kann, dass die weitere Entwicklung der ausgewogenen Kontrolle eines Standardisierungsprozesses unterliegt.

Es existieren verschiedene Dokumentstandards und Spezifikationen. Diese umfassen HTML, XHTML, PDF und die Teilmengen ODF, DocBook, DITA und RTF. Wie die zahlreichen Standards zur Darstellung von Bitmapbildern (einschließlich TIFF/IT, TIFF/EP, JPEG 2000 und PNG) wurden auch diese Formate für einen jeweils eigenen Zweck geschaffen. Open XML erfüllt die Anforderungen an einen Standard, der die Merkmale im vorhandenen Dokumentkorporus abdeckt. Unseres Wissens nach ist dies das einzige XML-Dokumentformat, das alle Features der binären Formate unterstützt.

Open XML definiert Formate für Textverarbeitungs-, Präsentations- und Tabellenkalkulationsdokumente. Jeder Dokumenttyp wird durch eine primäre Markupsprache definiert: WordprocessingML, PresentationML oder SpreadsheetML. Einbettungsmechanismen ermöglichen für jeden dieser drei Dokumenttypen die Integration von Materialien in die anderen primären sowie in einer Reihe von unterstützenden Markupsprachen.

Die Spezifikation enthält sowohl normatives Material (zur Definition von Open XML) und informatives Material (zur Förderung des Leserverständnisses, jedoch ohne Beschreibungen). Sie ist in Teile gegliedert, um die Anforderungen verschiedener Zielgruppen zu erfüllen.

Teil 1 – Grundlagen 165 Seiten	<ul style="list-style-type: none"> — Definiert das Vokabular, die Anmerkungskonventionen und Abkürzungen — Enthält eine Übersicht der drei primären und der unterstützenden Markupsprachen — Definiert die Bedingungen für Konformität und stellt die Interoperabilitätsrichtlinien bereit — Beschreibt die Einschränkungen innerhalb der Open Packaging Conventions, die für die einzelnen Dokumenttypen gelten
Teil 2 – Open Packaging Conventions 125 Seiten	<ul style="list-style-type: none"> — Definiert die Open Packaging Conventions (OPC). Jede Open XML-Datei enthält eine Sammlung von Byteströmen, die als Teile bezeichnet werden. Diese Teile werden zu einem Container kombiniert, der Paket genannt wird. Das Paketformat wird von den OPC definiert. — Beschreibt die empfohlene physische Implementierung der OPC, die das Dateiformat ZIP verwendet — Deklariert die XML-Schemas für das OPC als XML Schema Definitions (XSD) (2) in einem Anhang, der nur in elektronischer Form herausgegeben wird. Der Anhang enthält außerdem nicht normative Darstellungen der Schemas mithilfe von RELAX NG (ISO/IEC 19757-2) (3).
Teil 3 – Vorbereitung 466 Seiten	<ul style="list-style-type: none"> — Stellt die Merkmale der einzelnen Markupsprachen dar, liefert den Kontext sowie illustrierende Elemente mithilfe von Beispielen und Diagrammen. Dieser Teil ist informativ (nicht normativ). — Beschreibt die Komponente zur Speicherung individueller XML-Daten innerhalb des Pakets, um die Integration in Geschäftsdaten zu unterstützen
Teil 4 – Referenz zur Markupsprache 5756 Seiten	<ul style="list-style-type: none"> — Definiert jedes Element und Attribut, die Hierarchie der Beziehungen über- und untergeordneter Elemente sowie (nach Bedarf) weitere semantische Besonderheiten. Dieser Teil dient als Referenz, wenn umfassende Informationen zu einem Element oder Attribut erforderlich sind.

- Definiert die Möglichkeiten zur Speicherung individueller XML-Daten
 - Deklariert die XML-Schemas für die Markup Sprachen als XSD (2) in einem Anhang, der nur in elektronischer Form herausgegeben wird. Der Anhang drückt diese Schemas nicht normativ und mithilfe von RELAX NG (ISO/IEC 19757-2) (3) aus.
- Teil 5 –
 Markupkompatibilität und
 Erweiterbarkeit
 34 Seiten
- Beschreibt die Möglichkeiten zur Erweiterung von Open XML-Dokumenten
 - Definiert Elemente und Attribute, mit denen Anwendungen mit unterschiedlichen Erweiterungen interagieren können
 - Drückt die Erweiterbarkeitsregeln mithilfe von NVDL (ISO/IEC 19757-4) (4) aus

Zur Vereinfachung des Lesens und der Navigation in diesen Dokumenten verfügen die elektronischen Versionen über zahlreiche aktive Links. Vor allem Teil 4 bietet durchgehend Links zu über- und untergeordneten Elementen.

4 EIGENSCHAFTEN DES STANDARDS

Dieser Abschnitt dient als Vorbereitung auf den ersten Umgang mit Open XML, denn darin werden einige der allgemeinen Eigenschaften erläutert. Jeder Unterabschnitt beschreibt eine dieser Eigenschaften und verweist auf spezifische Features innerhalb von Open XML.

- „Interoperabilität“ beschreibt, inwiefern Open XML von proprietären Formaten, Features und Laufzeitumgebungen unabhängig ist, sodass Entwickler mehr Auswahl haben.
- „Internationalisierung“ erwähnt einige repräsentative Arten, in denen Open XML alle größeren Sprachgruppen unterstützt.
- „Leichte Übernahme durch Entwickler“, „Kompaktheit“ und „Modularität“ führen spezifische Arten auf, in denen Open XML praktische Hindernisse für die Implementierung durch verschiedene Parteien vermeidet oder eliminiert: Lernkurve, Mindestfeatures und Leistung.
- „Zuverlässige Migration“ beschreibt, wie Open XML das übergreifende Ziel des Informationserhalts erreicht (einschließlich des Interesses des ursprünglichen Autors an vorhandenen und neuen Dokumenten).
- „Integration in Geschäftsdaten“ erläutert, wie Open XML geschäftliche Daten in individuelle Schemas einbindet, um Integration und Wiederverwendung von Informationen zwischen Produktivitätsanwendungen und Informationssystemen zu ermöglichen.
- „Raum für Innovationen“ beschreibt, wie Open XML zukünftige Entwicklungen durch Definition neuer Erweiterungsmechanismen fördert und wie dieser Standard die Interoperabilität von Anwendungen mit verschiedenen Featuresätzen unterstützt.

Der Rest dieses Dokuments (einschließlich dieses Abschnitts) enthält eine thematische Übersicht zu Open XML. Verweise auf die Spezifikation erfolgen in der Form §Teil: Abschnitt.Unterabschnitt. Beispiel: §1:2.5 verweist auf Teil 1, Abschnitt 2, Unterabschnitt 5 der Spezifikation. Verweise auf andere Überschriften in diesem Dokument erfolgen durch Nennung der Überschrift.

4.1 INTEROPERABILITÄT

Entwickler können Anwendungen schreiben, die Open XML auf verschiedenen Plattformen nutzen oder erzeugen.

Die Interoperabilität von Open XML wurde vor allem durch umfassende Erweiterungen, Modifikationen und Überprüfungen der Spezifikation durch Mitglieder des Ecma TC45-Ausschusses (1) mit verschiedenen Hintergründen und Unternehmensinteressen ermöglicht. In diesem Ausschuss waren folgende Gruppen vertreten:

- Softwareanbieter (Apple, Intel, Microsoft, NextPage, Novell und Toshiba) mit mehreren Betriebssystemen (Linux, MacOS und Windows) und breit gefächerten Absichten zur Nutzung von Open XML
- Konzerne (BP, Barclays Capital, Essilor, Statoil) mit erheblichen Investitionen in vorhandene Inhalte, einschließlich geschäftskritischer Transaktionssysteme
- Die British Library und die United States Library of Congress mit unmittelbarem Interesse am Erhalt von Informationen

Im Zuge der Vorbereitung brachten die Ausschussmitglieder Hunderte von Problemen hinsichtlich Richtlinien, Klarheit, Semantik und möglichen Abhängigkeiten von der Umgebung zur Sprache.

Zu den repräsentativen Problemen und anderen Aktivitäten zählten folgende:

- Features zur Unterstützung der Plattformunabhängigkeit von Mechanismen, die in den ursprünglichen Binärformaten als proprietär galten
- Konformitätsbedingungen
- Inhalte der Schemas
- Alternative Darstellungen für die Schemas und Erweiterungsmechanismen mithilfe von RELAX NG (ISO/IEC 19757-2) und NVDL (ISO/IEC 19757-4) (4)
- Entwicklung von Tools zur automatischen Analyse und Visualisierung der Schemas
- Internationalisierung
- Vollständigkeit, Korrektheit und Klarheit der Beschreibungen in der gesamten Spezifikation, in vielen Fällen als Folge des Versuchs, Teile der Spezifikation zu implementieren

Im übrigen Teil dieses Unterabschnitts wird auf bestimmte Bereiche eingegangen, in denen Open XML aus Gründen der Interoperabilität vom ursprünglichen Binärformat abweicht.

Eine der zentralen Anforderungen an die Interoperabilität ist die Unabhängigkeit von einer bestimmten Form von Quellinhalt.

- Open XML enthält keinerlei Beschränkung auf einen bestimmten Bild-, Audio- oder Videotyp. Bilder können beispielsweise in den Formaten GIF, PNG, TIFF, PICT, JPEG oder in einem anderen Format vorliegen (§1:14.2.12).
- Integrierte Steuerelemente können in beliebiger Form vorliegen, z. B. als Java oder ActiveX (§1:15.2.8).
- Schriftartspezifikationen in WordprocessingML können Schriftkriterien und PANOSE-Informationen enthalten, um das Auffinden einer Ersatzschriftart zu unterstützen, wenn die ursprüngliche Schriftart nicht verfügbar ist (§3:2.10.5).

Darüber hinaus ist Open XML nicht von der Laufzeitumgebung der Anwendung abhängig, in der ein Dokument erstellt wurde.

- Das typische Beispiel trifft auf ein externes Steuerelement oder eine Anwendung zu, die ein Bild für einen Teil der Bildschirmoberfläche erstellt. Um zu verhindern, dass das Steuerelement oder die Anwendung nicht verfügbar ist oder in einer gegebenen Laufzeitumgebung nicht ausgeführt werden kann, ermöglicht die Dokumentdatei die Integration einer Bilddarstellung. Dieser Mechanismus existiert auch im älteren Binärformat.
- Open XML bietet einen allgemeineren Mechanismus namens Alternate Content Block (§3:2.18.4). Dieser kann in verschiedenen Situationen verwendet werden, in denen eine lesende Anwendung die Ausgabe einer erzeugenden Anwendung womöglich nicht interpretieren kann. Dieser Mechanismus wird in der Regel im Kontext der Erweiterbarkeit verwendet. Er wird im Unterabschnitt „Raum für Innovationen“ näher beschrieben.
- Open XML vermeidet die Abhängigkeit von allen Parametern, die zwar in der Umgebung des Dokumenterstellers, nicht jedoch in der Umgebung des Dokumentnutzers von Bedeutung sind. Der Parameter CT_SYSCOLOR ist beispielsweise ein Index für eine Farbtabelle in der Produktionsumgebung. Zur Unterstützung der Portierbarkeit in eine andere Nutzungsumgebung ermöglicht PresentationML dem Ersteller die Pufferung der Systemfarbe, die zum Zeitpunkt der Dokumenterstellung verwendet wurde.

Schließlich und vor allem entspricht Office Open XML den offenen W3C-Standards, wie z. B. XML (5) und XML-Namespaces (6). Allein diese Tatsache ermöglicht eine grundlegende Interoperabilität über alle Plattformen und Betriebssysteme hinweg, die diesen offenen Standards entsprechen.

4.2 INTERNATIONALISIERUNG

Open XML unterstützt Internationalisierungsfeatures, die aufgrund so unterschiedlicher Sprachen wie Arabisch, Chinesisch (drei Varianten), Hebräisch, Hindi, Japanisch, Koreanisch, Russisch und Türkisch erforderlich sind.

Open XML unterstützt selbstverständlich Unicode, da es sich um ein XML-Format handelt. Außerdem verfügt Open XML über umfassende Internationalisierungsfeatures, die im Laufe vieler Jahre optimiert wurden. Im Folgenden eine repräsentative Liste:

Textausrichtung: Open XML unterstützt Sprachen der Modelle Left-to-Right (LTR) und Right-to-Left (RTL). Außerdem werden bidirektionale Sprachen („BiDi“) wie Arabisch, Farsi, Urdu, Hebräisch und Jiddisch unterstützt, die von rechts nach links gelesen werden, jedoch Textsegmente mit anderer Ausrichtung enthalten können. In WordprocessingML kann die Textrichtung auf Absatzebene (§4:2.3.1.6) und auf Ebene eines Einzugs in einem Absatz (§4:2.3.2.28) gesteuert werden. In ähnlicher Weise kann Text in DrawingML auf Korpusebene (§4:5.1.5.1.1), auf Absatzebene (§4:5.1.5.2.2) und in nummerierten Aufzählungen (§4.5.1.5.4) gesteuert werden.

Textfluss: In WordprocessingML kann die Richtung des Textflusses auf Ebene eines Abschnitts oder einer Tabelle (§4:2.3.1.41) bzw. auf Absatzebene (§4:2.3.2.28) gesteuert werden. Auf Abschnitts- und Tabellenebene kann der Textfluss in vertikaler und horizontaler Richtung gesteuert werden. Auf diese Weise kann Open XML alle möglichen Textlayouts unterstützen (z. B. vertikale Zeilen von oben nach unten und von links nach rechts zur Unterstützung von Mongolisch). Dies beeinflusst die Layouts von Listen, Tabellen und anderen Präsentationselementen. DrawingML verwendet außerdem Kumimoji-Einstellungen auf Absatz- und Einzugebene, um Text horizontal und Zahlen vertikal fließen zu lassen (§4:5.1.5.2.3, §4:5.1.5.3.9). In WordprocessingML (§4:2.3.1.16) und PresentationML (§4:4.3.1.15) kann der Zeichenfluss auch mit Kinsoku-Einstellungen definiert werden, um festzulegen, mit welchen Zeichen eine Textzeile begonnen und beendet werden kann.

Zifferndarstellung: Zur Feldformatierung in WordprocessingML (§4:2.16.4.3), Absatz-/Listenummerierung in WordprocessingML (§4:2.9) und Nummerierung in DrawingML (§4:5.1.5.4, §4:5.1.12.61) können Zahlen mithilfe von mehreren Dutzend Zahlenformaten formatiert werden. Hierzu zählen Hiragana, Arabisch, Abjad, Thai, wörtliche Darstellung (z. B. „einhundertunddreißig“), Chinesisch, Koreanisch (Chosung oder Ganada), Hebräisch, Hindi, Japanisch, Lateinisch oder Vietnamesisch. Diese Möglichkeiten unterstützen auch die verschiedenen Dezimalwerte (z. B. „1.00“ und „1,00“) sowie Listentrennzeichen. Internationale Zahlenformate sind in SpreadsheetML besonders zuverlässig, da hier alle diese Features in Zellenformaten (§4:3.8.30) und in Verweisen auf externe Daten (§4.3.13.12) unterstützt werden.

Datumsangaben: In WordprocessingML (§4:2.18.7) und SpreadsheetML (§4:3.18.5) können Kalenderdaten in den Formaten Gregorianisch (drei Varianten), Hebräisch, Hijri, Japanisch (Kaiserzeit), Koreanisch (Tangun-Ära), Saka, Taiwanesisch und Thai geschrieben werden.

Formeln: Die Formelspezifikation in SpreadsheetML bietet verschiedene, auf die Internationalisierung bezogene Konvertierungsfunktionen, wie z. B. BAHTTEXT (§4:3.17.7.22), JIS (§4:3.17.7.185) und ASC (§4:3.17.7.11).

Sprachkennungen: In WordprocessingML (§4:2.3.2.18) und DrawingML (§4:5.1.5.3) kann jeder Absatz und Einzug mit einer Sprachkennung versehen werden, sodass die passenden Korrekturhilfen und andere sprachspezifische Funktionen von einer Anwendung ausgewählt werden können. Neben der Kennung für eine Sprache unterstützt Open XML die Benennung eines Zeichensatzes, einer Schriftfamilie und eines PANOSE-Werts, um die Anwendung bei der Auswahl einer passenden Ersatzschrift zu unterstützen, wenn keine lokale Schrift vorhanden ist.

4.3 LEICHTE ÜBERNAHME DURCH ENTWICKLER

Ein erfahrener Entwickler kann nach dem Lesen der Spezifikation schon nach kurzer Zeit mit dem Erstellen einfacher Open XML-Anwendungen beginnen.

Auch wenn die Spezifikation eine große Bandbreite an Features beschreibt, muss eine Open XML-konforme Anwendung nicht Unterstützung für alle Features in der Spezifikation bieten. Die Konformitätserklärung (§1:2) erfordert, dass ein konformer Nutzer „keine konformen Dokumente des [erwarteten] Dokumenttyps zurückweist“ und dass ein konformer Ersteller „konforme Dokumente erstellen kann“ (§1:2.5).

Außerdem bietet dieser Abschnitt Interoperabilitätsrichtlinien, die die Aufgaben der Elementsemantik definieren (§1:2.6).

Eine konforme Anwendung kann sehr spezifische Funktionen aufweisen. Es kann sich beispielsweise um eine Stapelverarbeitung handeln, die nur die Urheberrechtshinweise in verschiedenen Textverarbeitungsdokumenten aktualisiert, oder um ein Programm zur Umwandlung von Text in Sprache, das eine Folienpräsentation analysiert und deren Inhalt in Audio umwandelt, während der Anwender von Folie zu Folie navigiert. Dank der Struktur des Dateiformats können solche Programme mit minimalen Kenntnissen von Open XML geschrieben werden. Insbesondere gilt Folgendes:

- Das Dateiformat entspricht etablierten Standards, vor allem XML und ZIP, für die ausgereifte Tools existieren.
- Das Dateiformat nutzt die Open Packaging Conventions, die XML und ZIP mit Standardmechanismen kombinieren, um die Beziehungen innerhalb einer Datei auszudrücken. Aus diesem Grund kann der Inhalt einer Datei häufig ohne Kenntnisse der Kennzeichnungssemantik der primären oder unterstützenden Markupssprachen in Open XML durchlaufen werden.
- Elemente, die sich an untergeordneter Stelle in der XML-Struktur befinden, können ohne Beeinflussung der übrigen Struktur aufgerufen und modifiziert werden.

Kleine Details der Dateiformate, die zum Teil nicht in den Binärformaten enthalten waren, unterstützen Anwendungen mit minimaler Funktionalität, indem sie zwischengespeicherte Werte bereitstellen. Beispiel:

- Ohne Implementierung eines Seitenumbruchs kann eine Anwendung (z. B. ein Leseprogramm für Sehbehinderte) eine Seitennavigation anhand der zuletzt berechneten Seitenumbrüche anbieten (§4:2.3.3.13).
- Ohne Implementierung von Formeln und Integration in eine externe Datenquelle kann eine Tabellenkalkulation mit zwischengespeicherten Berechnungen (§3:3.2.9) und zwischengespeicherten externen Daten (§4:3.14 und §4:3.10.1.76) arbeiten.

Das minimal konforme Dokument ist sehr einfach aufgebaut; siehe hierzu den Unterabschnitt „Minimales WordprocessingML-Dokument“.

4.4 KOMPAKTHEIT

Das Open XML-Dateiformat unterstützt die Erstellung leistungsfähiger Anwendungen. In diesem Unterabschnitt beschreiben wir einige der Entwurfsaspekte, die zu einer kompakten Datei führen und somit die Handhabung und das Durchsuchen beschleunigen. Im nächsten Unterabschnitt zeigen wir, wie eine modulare Dateistruktur eine Anwendung in die Lage versetzt, viele Aufgaben durch Durchsuchen oder Ändern einer kleinen Teilmenge des Dokuments zu erledigen.

Eine Open XML-Datei wird in der Regel zur Paketerstellung und Komprimierung in einem ZIP-Archiv gespeichert. Diese Vorgehensweise basiert auf der in den Open Packaging Conventions empfohlenen Implementierung. Möglicherweise ist es für Sie überraschend, dass Open XML-Dateien durchschnittlich um 25 % und vereinzelt sogar bis zu 75 % kleiner als die binären Versionen sind. Dieses Whitepaper ist beispielsweise im Binärformat um 85 % größer!

Eine zweite einfache Ursache für die Kompaktheit (vor allem, wenn eine unkomprimierte Darstellung erforderlich ist) ist die Länge der XML-Kennungen. Häufig verwendete Kennzeichnungsnamen sind kurz. Für die Implementierung wird auch die Verwendung kurzer Namespacepräfixe empfohlen. Das herkömmliche Präfix für den WordprocessingML-Namespace lautet beispielsweise „w“.

Die Kompaktheit wird außerdem durch Vermeiden von Wiederholungen im Dateiformat gefördert. Eine Beispielklasse eliminiert die redundante Speicherung großer Objekte.

- In SpreadsheetML werden wiederholte Zeichenfolgen in einer Zeichenfolgentabelle der Arbeitsmappe abgelegt und durch einen Index referenziert (§3:3.3).
- In SpreadsheetML wird eine Formel, die vertikal und horizontal über mehrere Zellen verläuft, in der linken oberen Zelle als „Master-Formel“ abgelegt. Die übrigen Zellen im Bereich verweisen mit einem Gruppierungsindex auf diese Formel (§3:3.2.9.2).
- In DrawingML werden Formnamen (§4:5.1.12.56), Textgeometrien (§4:5.1.12.76) und andere Voreinstellungen (verschiedene Stellen in §3:5.8, §3:5.9 und §4:5.1.12) mit Namen oder Nummer angegeben anstatt ausdrücklich dargestellt. In diesen Fällen ist die Bedeutung der Namen und Nummern in der Spezifikation und nicht in der Datei enthalten. Hier ist die gewählte Darstellung das Ergebnis einer bewussten Kompromissentscheidung im Verlauf des Standardisierungsprozesses. Sie ist kompakt und ermöglicht die Bearbeitung auf der richtigen Abstraktionsebene: Ein Rechteck kann beispielsweise durch Ändern eines Attributs in ein Oval geändert werden (§4:5.1.11.18).

In einer anderen Beispielklasse dient die Hierarchie der Bereitstellung einer Vererbungssemantik. Als erfreuliches Nebenprodukt wird damit die Leistung gesteigert, da die Dateien kleiner ausfallen.

- In WordprocessingML sind Vorlagen hierarchisch (§3:2.8.9).
- In DrawingML werden Formen hierarchisch gruppiert (§4:5.1.2.1.20).
- In PresentationML verbindet eine Standardhierarchie Folienmaster, Folienlayouts und Folien (§3:4.2).

Weitere Aspekte von Open XML sind ebenfalls auf die Unterstützung der effizienten Implementierung ausgelegt. In SpreadsheetML speichert die Zellentabelle nur nicht leere Zellen und kann verbundene Zellen als eine Einheit anzeigen. Die Wirtschaftlichkeit dieser Methode ist ein signifikantes Merkmal schlanker Tabellen.

4.5 MODULARITÄT

Eine Anwendung kann viele Aufgaben erledigen, indem sie eine kleine Teilmenge des Dokuments durchsucht oder ändert.

Drei Features des Open XML-Formats greifen ineinander, um diese Modularität zu ermöglichen.

- Ein Dokument ist nicht monolithisch, sondern besteht aus mehreren Teilen.
- Die Beziehungen zwischen den Teilen werden in diesen gespeichert.
- Das ZIP-Archivformat, das in der Regel zur Unterstützung von Open XML-Dokumenten verwendet wird, unterstützt den zufälligen Zugriff auf alle Teile.

Beispiel:

- Eine Anwendung kann eine Folie problemlos von einer Präsentation in eine andere verlagern – und zwar zusammen mit Ressourcen wie Bildern und Layouts – ohne den Folieninhalt durchsuchen zu müssen (§3:13.3.8). Dieses Beispiel verwendet Daten, die als explizite Beziehung bezeichnet werden, um die Folie und deren Ressourcen zu finden. Explizite Beziehungen werden von den Open Packaging Conventions definiert und können ohne Kenntnis der PresentationML-Kennzeichnungssemantik durchsucht werden (§1:9.2, §2:8.3).
- Eine Anwendung kann alle Anmerkungen aus einem WordprocessingML-Dokument entfernen, ohne dessen Inhalt zu durchsuchen (§1:11.3.2). In diesem Beispiel werden zum Suchen nach den Anmerkungen Daten verwendet, die als implizite Beziehungen bezeichnet werden. Implizite Beziehungen sind Open XML-spezifisch und erfordern daher Kenntnisse der betreffenden Markupsprache (§1:9.2).

4.6 ZUVERLÄSSIGE MIGRATION

Open XML unterstützt alle Features der binären Formate von Microsoft Office 97-2003.

Die Probleme auf dem Weg zu diesem Ziel und die daraus folgende Einzigartigkeit von Open XML lassen sich nur schwer beschreiben. Einige Formate, wie z. B. PDF, bieten Endanwendern eine visuelle Kopie eines fertigen Dokuments. Im Gegensatz dazu unterstützt Open XML die zukünftige Bearbeitung oder Manipulation auf der Abstraktionsebene, die auch dem Autor zur Verfügung stand. Die Reduzierung einer Vektorgrafik zu einer Bitmap würde diese Möglichkeit beispielsweise nicht bieten. Gleiches gilt für die Reduzierung einer Vorlagenhierarchie zu unabhängigen Vorlagen. Darüber hinaus kann ein Dokument Semantiken für die Datenverarbeitung enthalten, die der ursprüngliche Autor beibehalten möchte. Hierzu zählen z. B. Formeln, die auf den Ergebnissen von Zwischenberechnungen basieren (einschließlich Fehlercodes oder Animationsregeln, die ein dynamisches Verhalten hervorrufen).

Diese Verweise auf die Spezifikation unterstreichen die Möglichkeiten von Open XML zur Darstellung subtiler Aspekte der Binärformate.

- Die SpreadsheetML-Beschreibung umfasst eine ausführliche Formelspezifikation (§4:3.17.7).
- Die WordprocessingML-Spezifikation dokumentiert die Regeln, nach denen Absatz-, Zeichen-, Nummerierungs- und Tabelleneigenschaften mit direkter Formatierung kombiniert werden (§3:2.8, insbesondere §3:2.8.10).
- Die PresentationML-Spezifikation dokumentiert die Animationsfeatures (§3:4.4).

Open XML ermöglicht die Konformität mehrerer Implementierungen, ohne dass diese in allen unbedeutenden Details übereinstimmen müssen. Dies ist vor allem dann von Bedeutung, wenn numerische Verarbeitungen anstehen, wie z. B. bei Layout, Effektrendering und FormelAuswertung. Wäre mehr Konsistenz als aus praktischen Gründen notwendig gefordert, entstünde für Entwickler eine unnötig hohe Hürde auf dem Weg zur Konformität. Diese Aussagen unterstreichen die Beispielentscheidungen, die in dieser Angelegenheit vom Ausschuss getroffen wurden.

- Open XML definiert Effekte, wie z. B. Oberflächenstrukturen (§5.1.12.50), ohne dass der Entwickler diese Effekte Pixel für Pixel angleichen muss.
- Open XML definiert Parameter wie Seitenränder (§4:2.6.11), Schriftarten (§4:2.8) und Ausrichtung (§4:2.3.1.13). Entwickler können verschiedene Flussalgorithmen implementieren, sofern sie diese Parameter berücksichtigen.

- Die SpreadsheetML-Formelspezifikation (§4:3.17.7) versucht nicht, Variationen in Fließkommaberechnungen zu eliminieren, da dies grundsätzlich voraussetzt, dass konforme Anwendungen langsame Emulationen implementieren, anstatt sich auf native Hardware zu verlassen. Vielmehr wird die Mindestanzahl von Präzisionsbits für numerische Berechnungen definiert (§4:3.17.5).
- Die SpreadsheetML-Formelspezifikation lässt außerdem bestimmte Fallentscheidungen in Abhängigkeit von der Implementierung zu, um zukünftige Innovationen zu unterstützen. Es wird beispielsweise nicht festgelegt, wie viele Iterationen einer Berechnung wie NORMINV (§4:3.17.7.227) zulässig sind. (Die Funktion NORMINV kehrt die normale Verteilung mithilfe einer iterativen Suche um.)

Verschiedene ältere Features, wie z. B. VML (§3:6), sind hauptsächlich aus Gründen der Abwärtskompatibilität enthalten. Für das Erstellen neuer Dokumente wird die Verwendung neuerer, bereits in Open XML enthaltener Standards, wie z. B. DrawingML (§3:5), empfohlen.

4.7 INTEGRATION IN GESCHÄFTSDATEN

Mithilfe von Open XML können Unternehmen unter Verwendung von benutzerdefinierten Schemas in Open XML-Dokumenten Produktivitätsanwendungen in Informationssysteme integrieren, die für die Geschäftsprozessverwaltung eingesetzt werden. Das Ziel eines Unternehmens bei dieser Herangehensweise besteht in der Wiederverwendung und Automatisierung der Verarbeitung von Geschäftsdaten, die sonst in Dokumenten verborgen sind, sodass sie von Unternehmensanwendungen nicht gelesen werden können und diese nicht in sie schreiben können.

Zu diesen Anwendungen zählen Folgende:

- *Suche*: Ein Endanwender kann eine Sammlung von Tabellenkalkulationen nach Unternehmen durchsuchen, deren Gewinnmarge 20 % übersteigt.
- *Metadatenkennzeichnung*: Ein Unternehmen kann alle Präsentationen kennzeichnen, die hinsichtlich bestimmter Richtlinien freigegeben wurden.
- *Dokumentzusammenstellung*: Eine Angebotsgruppe kann die Erstellung von Angeboten optimieren, indem die Vorbereitung der zugrunde liegenden Daten automatisiert wird.
- *Wiederverwendung von Daten*: Ein Vertriebsmitarbeiter kann einen Bericht zu allen Vertriebsverträgen innerhalb eines bestimmten Zeitraums erstellen und darin Kunden, Summen und veränderte Konditionen aufführen.
- *Anwendungen für Geschäftsbereiche*: Experten aus spezialisierten vertikalen Bereichen können Projektleistungen in einer vertrauten Erstellungsumgebung vorbereiten. Die Ergebnisse ihrer Arbeit fließen automatisch in Geschäftssysteme ein.

Zum Erreichen dieser Ziele müssen die Struktur und die Art der Daten definiert werden, die eine Dokumentklasse enthalten kann. Außerdem müssen die Informationen immer dann zur Verfügung stehen, wenn dies im Rahmen des Dokumentflusses erforderlich ist. Nehmen wir einen einfachen Lebenslauf als Beispiel. In diesem Fall wird eine Datenstruktur definiert, die die Felder „Name“, „Telefon“, „Adresse“, „Karriereziele“ und „Qualifikationen“ enthält. Diese Felder werden anschließend so definiert, dass diese von menschlichen Autoren in ein Dokument eingefügt werden können. In einer anderen geschäftlichen Situation (z. B. in einem Finanzkonzern oder einem medizinischen Zentrum) würden andere Strukturen und Datenfelder verwendet werden.

Open XML ermöglicht den Ablauf dieses Prozesses in standardisierter Form.

Zunächst wird die Struktur der Geschäftsdaten mit einem individuellen XML-Schema ausgedrückt. Auf diese Weise kann das Unternehmen Daten mit Kennungen ausdrücken, die aus geschäftlicher Sicht sinnvoll sind. Ein Unternehmen kann eigene Schemas erstellen oder branchenübliche Schemas verwenden, wie XBRL für Finanzberichte (7) und HL7 für Gesundheitsinformationen (8). Die Schemas werden im öffentlichen Bereich, in Unternehmen und als Branchenstandards erstellt. Ihr Zweck reicht von Geburtsurkunden bis hin zu Versicherungsinformationen. Sie können jedes individuelle Schema verwenden, sofern dieses in XSD-Form vorliegt (2).

Zweitens werden die individuellen Daten in einen individuellen XML-Teil (§3.7.3) eines Open XML-Dokuments eingebettet und können mit einem individuellen XML-Dateneigenschaftsteil (§4:7.5) beschrieben werden. Durch die Trennung dieser individuellen Daten von der Präsentation ermöglicht Open XML eine saubere Datenintegration sowie die Präsentation und Manipulation durch den Endanwender in verschiedenen Kontext-Szenarien, einschließlich Dokumenten, Formularen, Folien und Tabellenkalkulationen. Interoperabilität lässt sich daher auf einer grundlegenden und semantisch genauen Ebene erreichen.

4.8 RAUM FÜR INNOVATIONEN

Open XML soll Entwickler dazu animieren, neue Anwendungen zu erstellen, an die man zum Zeitpunkt der Definition binärer Formate oder sogar bei der Definition von Open XML noch nicht gedacht hat.

Zunächst befassen wir uns mit Erweiterungsmöglichkeiten, die in ihrer Gesamtheit die Interoperabilität zwischen Anwendungen mit abweichenden Featurekombinationen ermöglichen. Nehmen wir als Beispiel eine *übergeordnete* Anwendung (die ein neues Feature enthält, das noch nicht in Open XML dokumentiert ist) und eine *untergeordnete* Anwendung (die dieses Feature nicht kennt). Die drei primären Ziele der Erweiterbarkeit sind folgende:

- *Visuelle Originaltreue*: Die Möglichkeit der untergeordneten Anwendung, Informationen so wie die übergeordnete Anwendung anzuzeigen. Dies setzt zwangsläufig voraus, dass in einer Datei mehrere Darstellungen der gleichen Daten gespeichert werden.
- *Bearbeitbarkeit*: Die Möglichkeit zur Bearbeitung einer oder mehrerer dieser Darstellungen
- *Datenschutz*: Die Möglichkeit, sicherzustellen, dass alte Versionen einer Darstellung nach Bearbeiten einer anderen Darstellung nicht beibehalten werden, sodass keine Informationen zurückbleiben, von denen der Anwender glaubt, sie seien gelöscht oder geändert worden. Eine Anwendung kann dieses Ziel durch Eliminieren oder Synchronisieren von Darstellungen erreichen.

Ein Entwickler, der das Open XML-Feature erweitern möchte, hat vor allem zwei Optionen:

- *Alternative Inhaltsblöcke*: Ein alternativer Inhaltsblock (§3:2.18.4 und §5:9.2) speichert mehrere Darstellungen des gleichen Inhalts, jeweils in einem Block seiner Wahl. Eine untergeordnete Anwendung liest einen Block ihrer Wahl, den sie lesen kann. Bei der Bearbeitung werden so viele Blöcke geschrieben, wie diese Anwendung schreiben kann.
- *Erweiterungslisten*: Eine Erweiterungsliste (§3:2.6) speichert individuelle XML-Daten ohne visuelle Darstellung.

Entwickler können außerhalb dieser Erweiterbarkeitsmechanismen Innovationen anbringen.

- *Alternative Interaktionsparadigmen.* Open XML definiert mehr als eine Dokumentsyntax, jedoch weniger als ein Anwendungsverhalten. Wie in der Konformitätserklärung beschrieben, liegt der Schwerpunkt auf der Semantik (§1:2.2, §1.2.3). Daraus folgt, dass eine konforme Anwendung uneingeschränkt auf verschiedene Arten oder gar nicht mit einem Endanwender kommunizieren kann, sofern sie die definierte Semantik berücksichtigt.
- *Neuartige Datenverarbeitungsumgebungen.* Die Konformitätserklärung lässt Anwendungen mit geringer Kapazität zu, sodass diese auf kleinen Geräten ausgeführt werden können. Auch Anwendungen, die nur eine Teilmenge von Open XML implementieren, sind zulässig (§1:2.6). Der Mechanismus für weitere Merkmale ermöglicht einer produzierenden Anwendung die Mitteilung ihrer Kapazitätsgrenzen (§3:8.1).

Wie im vorherigen Unterabschnitt angegeben, ist das Rendern von Dokumenten für die direkte Anwenderinteraktion bei einigen der umfassendsten Innovationsmöglichkeiten nicht vorgesehen. Stattdessen umfassen sie die systeminterne Verarbeitung mit XML-Mitteilungsformaten, z. B. über XML Web-Services (9). Zwar weisen Anwendungen dieser Art bis auf die auf Daten in Open XML-Dokumenten basierenden Abläufe kein für die Benutzer sichtbares Verhalten auf, doch sie unterliegen der (rein syntaktischen) Dokumentkonformität (§1:2.4) und der Anwendungskonformität (§1:2.5) sowie den Interoperabilitätsrichtlinien (§1:2.6), die auch die Semantik umfassen.

Obwohl sich kaum alle möglichen Anwendungsfälle für individuelle XML-Verarbeitung aufzählen lassen, sind XML-orientierte Dienste denkbar, die Open XML-Dokumente zur automatischen Extraktion und Integration individueller Daten und Sicherheitsdienste, wie XML Digital Signature (10) oder XML-Verschlüsselung (11), verarbeiten. Auch weitere XSLT-Transformationen (12) zur Umwandlung von anderen/in andere XML-Formate sind denkbar. Open XML schränkt diese Verarbeitung in keiner Weise ein.

5 STRUKTUR EINES OFFICE OPEN XML-DOKUMENTS

Ein primäres Ziel dieses Whitepapers ist es, den Leser beim Verständnis der übergeordneten Struktur einer Open XML-Datei zu unterstützen. Hierzu bieten wir ein angemessenes Maß an Details zu den Open Packaging Conventions (OPC) und weniger detaillierte Informationen zu den einzelnen Markupsprachen.

5.1 OPEN PACKAGING CONVENTIONS

Die Open Packaging Conventions (OPC) bieten eine Möglichkeit zur Speicherung mehrerer Inhaltsarten (z. B. XML, Bilder und Metadaten) in einem Container, wie z. B. einem ZIP-Archiv, um eine vollständige Dokumentdarstellung zu erreichen. Sie beschreiben ein logisches Modell zur Darstellung von Inhalten und Beziehungen.

Die empfohlene Implementierung für die OPC verwendet das ZIP-Archivformat. Die Struktur einer Open XML-Datei kann mit einem ZIP-Anzeigeprogramm eingesehen werden. Es ist sinnvoll, den Inhalt einer kleinen Open XML-Datei während der Lektüre dieser Beschreibung auf diese Weise zu untersuchen. Unter Windows müssen Sie die entsprechende Datei einfach mit der Erweiterung ZIP versehen und mit einem Doppelklick öffnen.

Bei einem Open XML-Dokument handelt es sich logischerweise um ein OPC-Paket (§5:8). Ein Paket ist eine einfache Sammlung von *Teilen* (§5:8.1). Jeder Teil verfügt über einen schreibweisenabhängigen *Teilnamen*, der aus einer durch Schrägstrich getrennten Folge von Segmentnamen besteht. Beispiel: „/pres/slides/slide 1. xml“ (§5:8.1.1). Jeder Teil weist außerdem einen *Inhaltstyp* auf (§5:8.1.2). Das ZIP-Archiv besteht physisch aus einem Paket. Jedes ZIP-Element im Archiv ist ein Teil, und Pfadnamen innerhalb des Archivs sind eine direkte Entsprechung von Teilnamen.

In der ZIP-Implementierung ermöglicht „/[Inhaltstypen].xml“ einem Anwender die Ermittlung des Inhaltstyps jedes Teils im Paket (§2:9.2.6). Die Syntax und Definition der Medientypen entsprechen Abschnitt 3.7 von RFC 2616 (13).

Pakete und Teile können *explizite Beziehungen* (§1:9.2) zu anderen Teilen im Paket sowie zu externen Ressourcen enthalten. Jede explizite Beziehung verfügt über eine Beziehungs-ID, die dem Inhalt eines Teils einen Bezug ermöglicht. Über den Typ kann eine Anwendung die Verarbeitung bestimmen. Beziehungstypen werden mithilfe von URIs benannt, sodass nicht koordinierende Parteien sicher und problemlos neue Typen erstellen können.

Alle expliziten Beziehungen eines Quellpakets werden in einem *Beziehungsteil* gespeichert. Der Beziehungsteil für das Paket als Ganzes lautet „/_rels/.rels“. Der Beziehungsteil für den Teil „/a/b/c.xml“ lautet „/a/b/_rels/c.xml.rels“. Die Beziehungsteile (und in der ZIP-Implementierung der Teil für den Inhaltstyp) sind die einzigen besonders benannten Teile in einem Paket. Zum Öffnen eines Pakets muss eine Anwendung den Teil mit den Paketbeziehungen durchsuchen und den Beziehungen des entsprechenden Typs folgen.

Alle anderen Teile in einem Open XML-Dokument enthalten Open XML, individuelles XML oder andere Inhalte, wie z. B. Multimediaobjekte. Die Möglichkeit eines Teils zur Speicherung von individuellem XML ist ein besonders leistungsfähiger Mechanismus zur Integration geschäftlicher Daten und Metadaten.

5.2 WORDPROCESSINGML

Ein WordprocessingML-Dokument besteht aus einer Sammlung von Komponenten (§3:2.1). Diese Komponenten setzen sich wie folgt zusammen: Hauptdokument (§3:2.2), Glossardokument (§3:2.13), ein Subdokument (§3:2.18.2), eine Kopfzeile (§3:2.11.1), eine Fußzeile (§3:2.11.2), eine Anmerkung (§3:2.14.5), ein Rahmen, ein Textfeld (§3:2.18.1), eine Fußnote (§3:2.12.1) oder eine Endnote (§3:2.12.2).

Das Hauptdokument ist die einzige erforderliche Komponente. Sie ist das Ziel der Paketbeziehung vom Typ:

<http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument>

Ein typischer Pfad von Stamm zu Blatt in der XML-Struktur umfasst die folgenden XML-Elemente (§3:2.2):

- `document` – Das Stammelement des Hauptdokuments (§3:2.3)
- `body` – Körper (§3:2.7.1). Kann mehrere Absätze enthalten. Kann auch die in einem `sectPr`-Element definierten Eigenschaften enthalten.
- `p` – Absatz (§3:2.4.1). Kann einen oder mehrere Runs enthalten. Kann auch die in einem `pPr`-Element definierten Absatzigenschaften enthalten. Diese können wiederum standardmäßige Runeigenschaften (auch Zeicheneigenschaften genannt) enthalten, die in einem `rPr`-Element definiert sind (§3:2.4.4).

- r – Run (§3:2.4.2). Kann mehrere Arten von Runinhalten enthalten, hauptsächlich Textbereiche. Kann auch Runeigenschaften enthalten (rPr). Ein Run ist ein elementares Konzept in Open XML. Ein Run ist ein Textabschnitt mit identischen Eigenschaften. Er enthält keine weiteren Textmarkierungen. Wenn ein Satz beispielsweise die Worte „Dies sind **drei** Runs“ enthält, wird er durch mindestens drei Runs dargestellt: „Dies sind“, „**drei**“ und „Runs“. In dieser Hinsicht weicht Open XML erheblich von Formaten ab, die das beliebige Schachteln von Eigenschaften ermöglichen (z. B. HTML).
- t – Textbereich (§3:2.4.3.1). Enthält eine beliebige Menge an Zeichen ohne Formatierung, Zeilenumbrüche, Tabellen, Grafiken oder andere nicht textliche Informationen. Die Formatierung des Texts wird von den Run- und Absatzzeigenschaften übernommen. Dieses Element verwendet häufig das Attribut `xml: space="preserve"`.

In diesem Unterabschnitt haben wir uns mit der direkten Textformatierung durch Absatz- und Runeigenschaften befasst. Die direkte Formatierung erfolgt am Ende der Anwendungsreihenfolge, die auch Zeichen-, Nummerierungs- und Tabellenvorlagen sowie Dokumentstandards umfasst (§3:2.8.10). Diese Vorlagen sind in Vererbungshierarchien organisiert (§3:2.8.9).

Im Unterabschnitt „Minimales WordprocessingML-Dokument“ weiter unten ist ein vollständiges WordprocessingML-Dokument aufgeführt.

5.3 PRESENTATIONML

Ein PresentationML-Dokument wird durch einen Präsentationsteil beschrieben. Der Präsentationsteil ist das Ziel der Paketbeziehung von Typ:

<http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument>

Die Präsentation bezieht sich auf die folgenden primären Konstrukte (§3:4.2), die von oben nach unten in der Standardhierarchie aufgelistet werden:

- Folienmaster, Notizmaster und Handoutmaster (§3:4.2.2). Diese übernehmen die Eigenschaften der Präsentation.
- Folienlayouts (§3:4.2.5). Diese übernehmen die Eigenschaften des Folienmasters.
- Folien (§3:4.2.3) sowie Notizseiten (§3:4.2.4). Diese übernehmen die Eigenschaften von Folienlayouts und Notizmastern.

Jeder Master, jedes Layout und jede Folie wird in einem eigenen Teil gespeichert. Der Name des Teils wird im Beziehungsteil des Präsentationsteils definiert. Jeder der sechs Teile (mit Ausnahme der Präsentation) ist auf die gleiche Art strukturiert. Ein typischer Pfad von Stamm zu Blatt in der XML-Struktur umfasst die folgenden XML-Elemente (§3:2.2):

- `sld`, `sldLayout`, `sldMaster`, `notes`, `notesMaster` oder `handoutMaster` – Das Stammelement
- `csld` – Folie (§4:4.4.1.15). Kann DrawingML-Elemente (siehe die beiden nächsten Punkte) und andere strukturelle Elemente (siehe unten) enthalten
- `spTree` – Formstruktur (§4:4.4.1.42). Kann Gruppenformeneigenschaften in einem `grpSpPr`-Element (§4:4.4.1.20) und nicht visuelle Gruppenformeneigenschaften in einem `nvGrpSpPr`-Element enthalten (§4:4.4.1.28). Dieser Knoten und seine Verzweigungen sind DrawingML-Elemente. Wir führen hier einige DrawingML-Elemente auf, da diese in PresentationML eine wichtige Rolle spielen.
- `sp` – Form (§4:4.4.1.40). Kann Formeneigenschaften in einem `spPr`-Element (§4:4.4.1.41) und nicht visuelle Formeneigenschaften in einem `nvSpPr`-Element enthalten (§4:4.4.1.31).

Neben dem DrawingML-Forminhalt kann ein `cslid`-Element andere strukturelle Elemente enthalten, je nach dem Stammelement, in dem es enthalten ist). Siehe hierzu die folgende Tabelle:

	Folie	Folien- layout	Folien- master	Handout- master	Notiz- master	Notiz- seite
Allgemeine Daten	X	X	X	X	X	X
Übergang	X	X	X			
Zeitsteuerung	X	X	X			
Kopf-/Fußzeilen		X	X	X	X	
Entsprechender Name		X				
Layout-Typ		X				
Erhaltung		X	X			
Layout-Liste			X			
Textvorlage			X			

Die von Objekten weiter unten in der Standardhierarchie (Folienmaster, Folienlayout, Folie) angegebenen Eigenschaften haben Vorrang vor den entsprechenden Eigenschaften weiter oben in der Hierarchie. Wird für eine Folie beispielsweise kein Übergang angegeben, wird dieser dem Folienlayout entnommen. Ist er dort nicht definiert, wird er dem Folienmaster entnommen.

5.4 SPREADSHEETML

Ein SpreadsheetML-Dokument wird auf oberster Ebene von einem Arbeitsmappenteil beschrieben. Der Arbeitsmappenteil ist das Ziel der Paketbeziehung mit folgendem Typ:

<http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument>

Der Arbeitsmappenteil speichert Informationen zur Arbeitsmappe und zu seiner Struktur, wie z. B. Dateiversion, Ausgangsanwendung und Kennwort für Änderungen. Die Arbeitsmappe enthält logischerweise eines oder mehrere Blätter (§3:3.2). Jedes Blatt wird physisch in einem eigenen Teil gespeichert und in üblicher Weise vom Arbeitsmappenteil referenziert. Jedes Blatt kann ein Arbeitsblatt, ein Diagrammblatt oder ein Dialogblatt sein. Wir befassen uns hier nur mit dem Arbeitsblatt (dem gängigsten Typ). In einem Arbeitsblattobjekt umfasst ein typischer Pfad von Stamm zu Blatt in der XML-Struktur die folgenden XML-Elemente:

- `worksheet` – Das Stammelement in einem Arbeitsblatt (§3:3.2)
- `sheetData` – Die Zellentabelle mit allen nicht leeren Zellen im Arbeitsblatt (§3:3.2.4)
- `row` – Eine Zellenreihe in der Zellentabelle (§3:2.8)
- `c` – Eine Zelle (§3:3.2.9). Das Attribut `r` gibt die Zellenposition mithilfe der A1-Koordinaten an. Die Zelle kann auch eine Vorlagenkennung (Attribut `s`) und einen Datentyp (Attribut `t`) aufweisen.
- `v` und `f` – Der Wert (§3:3.2.9.1) und eine optionale Formel (§3:3.2.9.2) der Zelle. Weist die Zelle eine Formel auf, ist der Wert das Ergebnis der letzten Berechnung.

Zeichenfolgen und Formeln werden in gemeinsamen Tabellen gespeichert (§3:3.3 und §3:3.2.9.2.1), um eine redundante Speicherung zu vermeiden und Lade-/Speichervorgänge zu beschleunigen.

5.5 UNTERSTÜTZENDE MARKUPSPRACHEN

Zur Beschreibung des Inhalts eines Open XML-Dokuments können verschiedene unterstützende Markupsprachen verwendet werden.

- DrawingML (§3:5) – Zur Darstellung von Formen und anderen grafisch erstellten Objekten in einem Dokument.
- VML (§3:6) – Ein Format für Vektorgrafiken, das aus Gründen der Abwärtskompatibilität enthalten ist und das zu gegebener Zeit durch DrawingML ersetzt wird.
- Gemeinsame MLs: Math (§3:7.1), Metadata (§3:7.2), Custom XML (§3:7.3) und Bibliography (§3:7.4).

5.6 MINIMALES WORDPROCESSINGML-DOKUMENT

Dieser Abschnitt enthält ein minimales WordprocessingML-Dokument, das aus drei Teilen besteht.

Der Inhaltstyp-Teil „/[Inhaltstypen].xml“ beschreibt die Inhaltstypen der beiden anderen erforderlichen Teile.

```
<Types xmlns="Error! Hyperlink reference not valid.">
  <Default Extension="rels"
    ContentType="application/vnd. Open XML formats -package. relationships+xml"/>
  <Default Extension="xml"
    ContentType="application/vnd.Open XMLformats-officedocument.wordprocessingml.document.main+xml"/>
</Types>
```

Der Paketbeziehungsteil „/_rels/.rels“ beschreibt die Beziehungen zwischen dem Paket und dem Hauptdokumentteil.

```
<Relationships xmlns="http://schemas.openxmlformats.org/package/2006/relationships">
  <Relationship Id="rId1"
    Type="http://schemas.openxmlformats.org/officeDocument/2006/relationships/officeDocument"
    Target="document.xml"/>
</Relationships>
```

Der Dokumentteil, in diesem Fall „/document.xml“, enthält den Dokumentinhalt.

```
<w:document xmlns:w="http://schemas.Open XMLformats.org/wordprocessingml/2006/main">
  <w:body>
    <w:p>
      <w:r>
        <w:t>Hello, world.</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

Die Spezifikation bietet minimale Dokumente und weitere Informationen zu WordprocessingML (§1:11.2), PresentationML (§1:13.2) und SpreadsheetML (§1:12.2).

Open XML ist das Produkt erheblicher Anstrengungen der Vertreter zahlreicher industrieller und öffentlicher Einrichtungen mit verschiedenen Hintergründen und organisatorischen Interessen. Es umfasst alle Features, die im vorhandenen Dokumentkorporus eingesetzt werden, sowie alle Internationalisierungsanforderungen aller größeren Sprachgruppen auf der ganzen Welt. Als Ergebnis der Standardisierungsarbeiten von Ecma TC45 (1) und Beiträgen aus öffentlichen Anmerkungen hat Open XML ein hohes Maß an Interoperabilität und Plattformunabhängigkeit ermöglicht. Seine Dokumentation ist (dank umfassendem Referenzmaterial) vollständig und (durch nicht normative Beschreibungen) verfügbar. Außerdem enthält dieser Standard ausreichende Informationen für die problemlose Verarbeitung von Dokumenten durch Hilfstechnologien. Open XML-Implementierungen können sehr klein sein und bieten spezifische Funktionalität. Sie können jedoch auch alle Features umfassen. In das Format integrierte Erweiterbarkeitsmechanismen bieten ausreichend Raum für Innovationen.

Die Standardisierung der Formatspezifikation und deren Pflege stellen sicher, dass mehrere Parteien diesen Standard sicher einsetzen und darauf vertrauen können, dass zukünftige Entwicklungen von den Prüfungen und Erwägungen eines auf offenen Standards basierenden Verfahrens profitieren. Es existiert ein dringender Bedarf für einen offenen Dokumentstandard, der zum Erhalt von Milliarden von Dokumenten beitragen kann, die in den bisherigen Binärformaten erstellt wurden, und gleichzeitig auch die Milliarden von Dokumenten sichern kann, die Jahr für Jahr hinzukommen. Technologische Fortschritte bei Hardware, Netzwerken und auf Standards basierenden Softwareinfrastrukturen machen dies möglich. Aufgrund der explosionsartigen Diversifizierung der Marktnachfrage (einschließlich erheblicher bisheriger Investitionen in geschäftskritische Systeme) wird dies zu einer vordringlichen Aufgabe.

1. **Ecma International.** TC45 – Office Open XML-Formate. *Ecma International*. [Online] <http://www.ecma-international.org/memento/TC45.htm>
2. **W3C.** XML-Schema. *World Wide Web Consortium*. [Online] <http://www.w3.org/XML/Schema>
3. **ISO.** ISO/IEC 19757-2:2003. *International Organization for Standardization*. [Online] <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=37605&ICS1=35&ICS2=240&ICS3=30>
4. **ISO.** ISO/IEC 19757-4:2006. *International Organization for Standardization*. [Online] <http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=38615&ICS1=35&ICS2=240&ICS3=30>
5. **W3C.** Extensible Markup Language (XML) 1.0 (vierte Ausgabe). *World Wide Web Consortium*. [Online] 2006. <http://www.w3.org/TR/2006/REC-xml-20060816/>
6. **W3C.** Namespaces in XML 1.0 (zweite Ausgabe). *World Wide Web Consortium*. [Online] 2006. <http://www.w3.org/TR/2006/REC-xml-names-20060816/>
7. **XBRL International.** XBRL-Spezifikationen. *Extensible Business Reporting Language*. [Online] <http://www.xbrl.org/Specifications/>

8. **Health Level Seven.** HL7 ANSI-Standards. *Health Level Seven*. [Online]
http://www.hl7.org/about/directories.cfm?framepage=/documentcenter/public/faq/ansi_approved.htm
9. **W3C.** W3C Web Services-Architektur. *World Wide Web Consortium*. [Online] 2002.
<http://www.w3.org/2002/ws/>
10. **W3C.** W3C XML-Signatur. *World Wide Web Consortium*. [Online]
<http://www.w3.org/Signature/>
11. **W3C.** W3C XML-Verschlüsselung. *World Wide Web Consortium*. [Online] 2001.
<http://www.w3.org/Encryption/2001/>
12. **W3C.** XSL und XSLT. *World Wide Web Consortium*. [Online]
<http://www.w3.org/Style/XSL/>
13. **W3C.** Hypertext Transfer Protocol – HTTP/1.1. *World Wide Web Consortium*. [Online]
<http://www.w3.org/Protocols/rfc2616/rfc2616.html>