



Visão Geral

Visão Geral do Office Open XML

ECMA TC45
TOM NGO (NEXTPAGE), EDITOR



1. Introdução

O Office Open XML (OpenXML) é um padrão aberto proposto para documentos produzidos em processadores de texto, apresentações e planilhas, que pode ser livremente implementado em diversas plataformas e aplicativos. A sua publicação beneficia organizações que pretendem implementar aplicativos que usem o formato, entidades públicas e privadas que comprem esse tipo de software, profissionais do ensino ou escritores. Por fim, todos os usuários desfrutarão os benefícios do padrão XML nos documentos, como estabilidade, preservação de recursos existentes, interoperabilidade e evolução contínua.

A padronização do OpenXML foi realizada pelo Technical Committee 45 (TC45) da Ecma International, formado por representantes da Apple, Barclays Capital, BP, The British Library, Essilor, Intel, Microsoft, NextPage, Novell, Statoil, Toshiba e Biblioteca do Congresso dos Estados Unidos (1).

Este artigo apresenta um resumo do OpenXML, para que o leitor:

- Compreenda as finalidades do OpenXML e a estrutura da sua especificação;
- Conheça as suas propriedades, como resolver compatibilidade com versões anteriores, preservação de recursos existentes, extensibilidade, esquemas de personalização, subconjuntos, multiplataformas, internacionalização e acessibilidade;
- Aprenda como percorrer a estrutura de alto nível de um arquivo OpenXML e navegar rapidamente para qualquer parte da especificação para obter mais detalhes.

2. Finalidades do Padrão

O OpenXML foi inteiramente projetado para representar com fidelidade o corpus preexistente de documentos de processadores de texto, apresentações e planilhas, codificados em formatos binários definidos pela Microsoft Corporation. O processo de padronização consistiu em incorporar de forma precisa, no XML, as capacidades necessárias para representar o corpus existente e ampliá-lo, oferecer documentação detalhada e viabilizar a interoperabilidade. Até o momento da elaboração do padrão, mais de 400 milhões de usuários já haviam produzido documentos em formatos binários, perfazendo uma cifra estimada de 40 bilhões de documentos, além dos outros bilhões gerados a cada ano.

Os formatos binários originais desses arquivos foram criados em uma época em que o espaço era precioso e o tempo de análise do código impactava substancialmente a experiência do usuário. Eles se baseavam na serialização direta das estruturas de dados residentes em memória, utilizadas pelos aplicativos do Microsoft® Office. As infra-estruturas modernas de redes, hardware e padrões (em especial o XML) permitem um novo projeto que favorece a implementação por vários fornecedores em diversas plataformas e, em outras palavras, abrem as portas para a evolução.

Paralelamente aos avanços tecnológicos, os mercados diversificaram-se e passaram a incluir uma nova gama de aplicações não contempladas originalmente pelo universo simples dos programas de edição de documentos. Essas novas aplicações:

- Automaticamente geram documentos de dados empresariais;
- Extraem dados empresariais de documentos e alimentam aplicações empresariais com eles;
- Executam tarefas restritas a uma pequena subseção do documento, preservando, ainda, a possibilidade de edição;
- Fornecem acesso a populações de usuários com necessidades especiais, como pessoas com deficiências visuais;
- São executadas em uma variedade de hardware, como os dispositivos móveis.

Talvez a questão mais importante seja a preservação de longa duração. Embora tenhamos aprendido a criar quantidades de informações exponencialmente maiores, a sua codificação foi e ainda é feita por meio de representações digitais vinculadas aos programas que as produziram que, uma ou duas décadas mais tarde, costumam tornar-se extremamente difíceis de ser compreendidas sem que haja perdas significativas. A preservação do investimento financeiro e intelectual nesses documentos (os existentes e os novos) tornou-se uma prioridade premente.

O surgimento desses quatro fatores – quais sejam, a adoção muito ampla de formatos binários, os avanços tecnológicos, as forças de mercado que demandam aplicações diversificadas e a dificuldade cada vez maior de uma preservação de longa duração – criaram um imperativo: definir um formato XML aberto e migrar bilhões de documentos com um mínimo de perda possível. Além disso, a padronização do formato XML aberto e a sua manutenção criam um ambiente no qual qualquer organização possa depender com segurança da estabilidade contínua da especificação e ter certeza de que as evoluções subsequentes desfrutem a segurança e o equilíbrio resultantes do processo de definição de padrões.

Existem diversos padrões e especificações para documentos: HTML, XHTML, PDF e seus subconjuntos, ODF, DocBook, DITA e RTF. Assim como os inúmeros padrões usados para representar imagens em bitmap (por exemplo, TIFF/IT, TIFF/EP, JPEG 2000 e PNG), cada um foi criado para atender a um conjunto específico de finalidades. O OpenXML responde à necessidade de criar um padrão que abranja os recursos existentes no corpus atual de documentos. Até onde nos foi possível verificar, esse é o único formato de documento XML compatível com todos os recursos de formatos binários.

3. Estrutura do Padrão

O OpenXML define formatos para documentos de processadores de texto, apresentações e planilhas. Cada tipo de documento é especificado por meio de uma linguagem de marcação primária: WordprocessingML, PresentationML ou SpreadsheetML. Mecanismos incorporados permitem que um documento de qualquer um desses três tipos contenha material codificado nas linguagens de marcação primárias e outras linguagens de marcação compatíveis.

A especificação contém materiais normativos (que definem o OpenXML) e informativos (para ajudar a compreensão pelo leitor) e está organizada em diferentes partes, para atender às necessidades de públicos variados.

Parte 1 - Noções básicas 165 páginas	<ul style="list-style-type: none">– Definição de vocabulário, convenções notacionais e abreviações.– Resumo das três linguagens de marcação primárias e das linguagens de marcação compatíveis.– Estabelecimento das condições para conformidade e orientações para interoperabilidade.– Descrição das restrições das convenções de pacotes abertos (Open Packaging Conventions - OPC) válidas para cada tipo de documento.
Parte 2 - Convenções para pacotes abertos 125 páginas	<ul style="list-style-type: none">– Definição das convenções para pacotes abertos (OPC). Todos os arquivos OpenXML consistem numa coleção de seqüências de bytes, as "partes", reunidas em um recipiente denominado "pacote". O formato do pacote é definido pelas OPC.– Descrição da implementação física recomendada pelas OPC, que utilizam o formato de arquivo zip.– Declaração dos esquemas XML das OPC, na forma de definições de esquemas XML (XSD) (2), em anexo distribuído em formato eletrônico. O anexo também inclui representações não-normativas dos esquemas, usando o RELAX NG (ISO/IEC 19757-2) (3).
Parte 3 - Informações preliminares 466 páginas	<ul style="list-style-type: none">– Introdução aos recursos de cada linguagem de marcação, com o contexto e os elementos ilustrados com exemplos e diagramas. Esta parte é informativa (não-normativa).– Descrição dos recursos de armazenamento personalizado de dados XML em um pacote, para a integração compatível com dados empresariais.
Parte 4 - Referência da linguagem de marcação 5.756 páginas	<ul style="list-style-type: none">– Definição de todos os elementos e atributos, hierarquia de relações entre elementos pai/filho e elementos adicionais de semântica, se necessário. Esta parte deve ser usada como referência sempre que for preciso obter detalhes completos sobre um elemento ou atributo.– Definição dos recursos de armazenamento de dados XML personalizados.– Declaração dos esquemas XML das linguagens de marcação, na forma de XSD (2), em um anexo distribuído em formato eletrônico. O anexo também inclui as representações não normativas dos esquemas, usando RELAX NG (ISO/IEC 19757-2) (3).
Parte 5 - Compatibilidade e extensibilidade de marcação 34 páginas	<ul style="list-style-type: none">– Descrição dos recursos de extensão de documentos OpenXML.– Especificação dos elementos e atributos responsáveis pela interoperação de aplicativos com extensões diferentes.– Definição das regras de extensibilidade, usando o NVDL (ISO/IEC 19757-4) (4).

Para facilitar a leitura e a navegação entre esses documentos, as versões eletrônicas oferecem diversos vínculos internos. Em especial, a Parte 4 é toda formada por vínculos para elementos pai e filho.

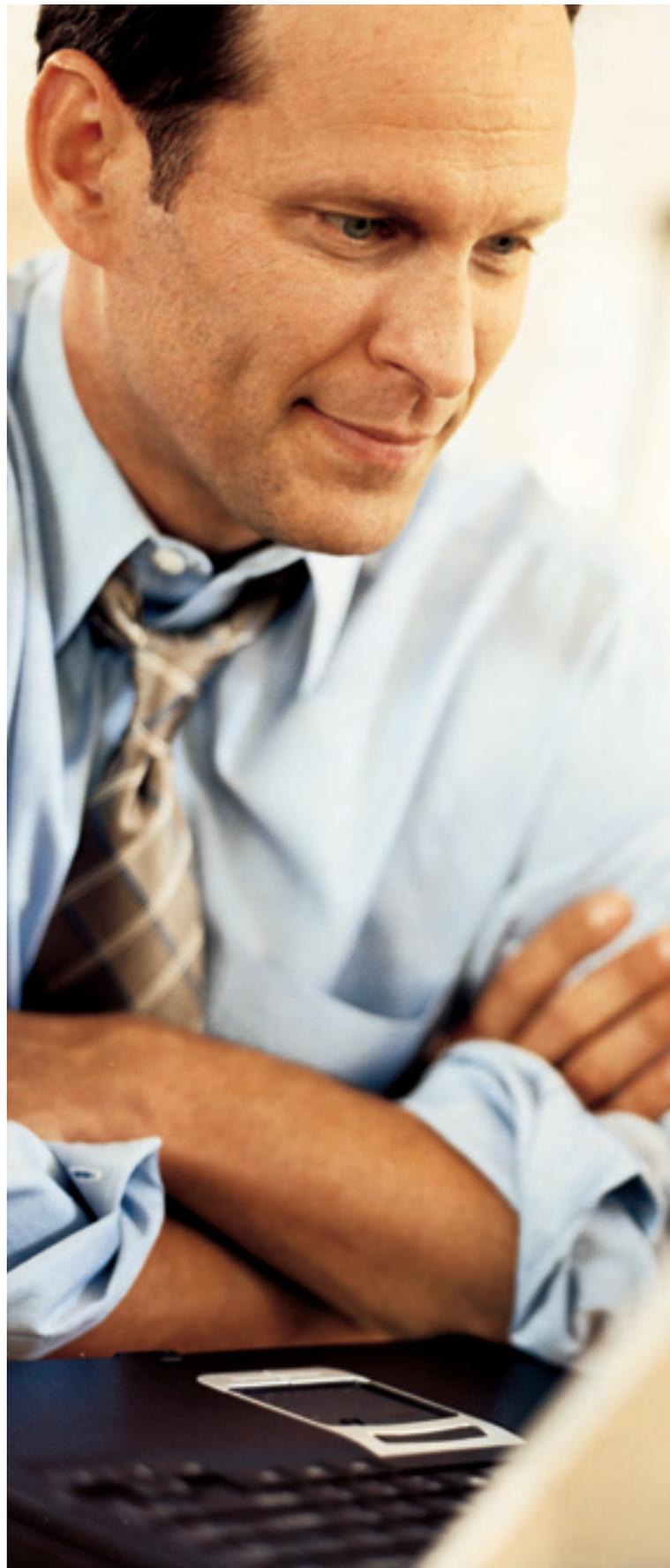
Acesse esses documentos no site <http://openxmldeveloper.org/default.aspx>

4. Propriedades do Padrão

Esta seção descreve algumas das propriedades de alto nível do OpenXML e permite que o leitor possa analisá-las. Cada subseção descreve uma dessas propriedades e menciona recursos específicos do OpenXML.

- **Interoperabilidade:** descreve como o OpenXML independe de formatos, recursos e ambientes de execução (runtime) proprietários, deixando aos desenvolvedores uma ampla gama de opções.
- **Internacionalização:** descreve algumas maneiras representativas pelas quais o OpenXML é compatível com cada grupo principal de linguagem.
- **Facilidade de adoção pelo desenvolvedor, Compacidade e Modularidade:** listam maneiras específicas pelas quais o OpenXML evita ou elimina os impedimentos práticos à implementação por diversos públicos-alvo: curva de aprendizagem, conjunto mínimo de recursos e desempenho.
- **Migração de alta fidelidade:** descreve como o OpenXML cumpre a vasta meta de preservar as informações, incluindo a intenção original do autor, nos documentos existentes e novos.
- **Integração com dados empresariais:** descreve como o OpenXML incorpora as informações empresariais em esquemas personalizados, para permitir a sua integração e reutilização entre aplicativos de produtividade e sistemas de informações.
- **Espaço para inovação:** descreve como o OpenXML prepara-se para o futuro pela definição de mecanismos adicionais de extensibilidade e a disponibilização da interoperabilidade aos aplicativos com conjuntos de recursos diferentes.

O restante deste documento, incluindo esta seção, é um guia organizado em tópicos para o OpenXML. As referências à especificação estão todas no formato \$parte:seção.subseção. Por exemplo: §1:2.5 refere-se à Parte 1, Seção 2.5 da especificação. As referências a outras porções deste documento são feitas pelo título respectivo.





4.1 Interoperabilidade

Os desenvolvedores podem escrever aplicativos que consomem e produzem OpenXML em diversas plataformas.

Acima de tudo, a interoperabilidade do OpenXML foi alcançada graças a contribuições, modificações e revisões extensivas da especificação por membros do comitê Ecma TC45 (1), com formações e interesses empresariais bastante variados. Representantes envolvidos:

- Fornecedores (Apple, Intel, Microsoft, NextPage, Novell e Toshiba) que adotam diferentes sistemas operacionais (Linux, MacOS e Windows) e têm diversos usos pretendidos para o OpenXML; corporações (BP, Barclays Capital, Essilor, Statoil) com investimentos pesados em conteúdos existentes, incluindo sistemas de transações de missão crítica;
- Biblioteca Britânica e Biblioteca do Congresso dos Estados Unidos, ambas com interesses diretos na preservação dos recursos.

Durante a preparação, os membros do comitê levantaram e resolveram centenas de questões referentes a política, clareza, semântica e possíveis dependências do ambiente. Os grupos de questões e atividades mais significativos foram:

- Recursos de apoio à independência de plataforma em relação a mecanismos que eram proprietários nos formatos binários originais;
- Condições para a conformidade;
- Conteúdo dos esquemas;
- Representações alternativas de esquemas e mecanismos de extensibilidade, usando o RELAX NG (ISO/IEC 19757-2) e o NVDL (ISO/IEC 19757-4) (4);
- Desenvolvimento de ferramentas para análise e visualização automáticas dos esquemas;
- Internacionalização;
- Completude, correção e clareza das descrições em toda a especificação, em muitos casos como resultado dos esforços de implementá-la parcialmente.

O restante desta subseção destaca áreas específicas nas quais o OpenXML desvia-se dos formatos binários originais, para favorecer a interoperabilidade. Um dos requisitos centrais da interoperabilidade é a independência de qualquer tipo de conteúdo de origem.

- O OpenXML não tem restrições quanto a tipos de imagem, áudio ou vídeo. Por exemplo, as imagens podem ser GIF, PNG, TIFF, PICT, JPEG ou de quaisquer outros tipos (§1:14.2.12).
-

- Os controles incorporados podem ser de qualquer tipo, como Java ou ActiveX (§1:15.2.8).
- As especificações de fonte do WordprocessingML podem incluir medidas de fonte e informações PANOSE, o que ajuda a encontrar uma substituição quando a fonte original não estiver disponível (§3:2.10.5).

Além disso, o OpenXML evita a dependência do ambiente de execução do aplicativo que produziu um documento.

- O exemplo clássico ocorre com um controle ou aplicativo externo que gere uma imagem parcial da superfície da exibição. Para prevenir o caso de o controle ou aplicativo estar indisponível ou não puder ser executado em um determinado ambiente, o arquivo do documento pode conter uma representação da imagem. Esse mecanismo existe também nos formatos binários mais antigos.
- O OpenXML introduz um mecanismo mais geral batizado como “bloco de conteúdo alternativo” (Alternate Content Block) (§3:2.18.4), que pode ser usado nas várias situações em que o aplicativo de consumo pode não ser capaz de interpretar o que um aplicativo de produção escreveu. É comumente usado no contexto da extensibilidade. O mecanismo é descrito em mais detalhes na subseção “Espaço para inovação”.
- O OpenXML elimina a dependência de qualquer parâmetro que seja significativo no ambiente de um produtor de documentos, mas não no ambiente do consumidor. Por exemplo, o parâmetro CT_SYSCOLOR é um índice interno de tabela de cores no ambiente de produção. Para tornar possível a portabilidade em um ambiente consumidor diferente, o PresentationML permite que o produtor armazene em cache a cor do sistema usada no momento da criação de um documento.

Por fim e mais importante: o Office OpenXML é compatível com padrões W3C, como o XML (5) e o XML Namespaces (6). Só esse fato já permite um nível básico de interoperabilidade em todas as plataformas e sistemas operacionais que adotam esses padrões abertos.

4.2 Internacionalização

O OpenXML oferece recursos de internacionalização demandados por idiomas tão dispares quanto árabe, chinês (três variantes), hebraico, hindi, japonês, coreano, russo e turco. O OpenXML é inerentemente compatível com o Unicode por ser XML. Além disso, o OpenXML oferece um conjunto rico de recursos de internacionalização aperfeiçoados ao longo de muitos anos. A lista é significativa:

Orientação do texto: o OpenXML é compatível com línguas da direita para a esquerda (DE) e da esquerda para a direita (ED). Também é compatível com línguas bidirecionais (“BiDi”) como árabe, farsi, urdu, hebraico e iídiche, cuja orientação é da direita para a esquerda, mas podem ter segmentos incorporados da esquerda para a direita. No WordprocessingML, a direção do texto pode ser controlada tanto no nível de parágrafo (§4:2.3.1.6) quanto no nível de “carreira” (segmentos de texto com propriedades idênticas), interna ao parágrafo (§4:2.3.2.28). Da mesma forma, no DrawingML, o texto pode ser controlado nos níveis de corpo (§4:5.1.5.1.1), parágrafo (§4:5.1.5.2.2) e lista numerada (§4.5.1.5.4).

Fluxo do texto: no WordprocessingML, a direção do texto pode ser controlada no nível de seção ou tabela (§4:2.3.1.41), ou de parágrafo (§4:2.3.2.28). Nos níveis de seção e tabela, o fluxo do texto pode ser controlado nas direções vertical e horizontal. Isso permite que o OpenXML seja compatível com todos os possíveis layouts de texto (como linhas verticais que estendem-se de cima para baixo e empilhadas da esquerda para a direita, para poder representar o idioma mongol). Isto afeta o layout de listas, tabelas e outros elementos de apresentação. O DrawingML também utiliza a configuração kumimoji nos níveis de parágrafo e carreira, para permitir a progressão horizontal do texto e a vertical de números (§4:5.1.5.2.3, §4:5.1.5.3.9). No WordprocessingML (§4:2.3.1.16) e no PresentationML (§4:4.3.1.15), o fluxo de caracteres também pode ser especificado pela utilização da configuração kinsoku, para especificar quais caracteres são permitidos para o início e o término de uma linha de texto.

Representação de números: para a formatação de campos no WordprocessingML (§4:2.16.4.3), numeração de parágrafos/listas no WordprocessingML (§4:2.9) e numeração no DrawingML (§4:5.1.5.4, §4:5.1.12.61), os números podem ser formatados de várias dezenas de possibilidades, incluindo hiragana, árabe, abjad, tailandês, texto cardinal (números escritos por extenso, como “cento e vinte três”), chinês, coreano (chosung ou ganada), hebraico, hindi, japonês, romano ou vietnamita.

Esses recursos também são compatíveis com valores decimais e de base de numeração (por exemplo, “1.00” vs. “1,00”) e separadores de lista arbitrários. A formatação internacionalizada de números é particularmente robusta no SpreadsheetML, compatível com todos os recursos dos formatos de células (§4:3.8.30) e referências a dados externos (§4.3.13.12).

Representação de datas: no WordprocessingML (§4:2.18.7) e no SpreadsheetML (§4:3.18.5), as datas de calendário podem ser escritas nos formatos gregoriano (três variantes), hebraico, hijri, japonês (era imperial), coreano (era Tangun), saka, mandarim e tailandês.

Fórmulas: a especificação de fórmulas no SpreadsheetML oferece diversas funções de conversão relacionadas à internacionalização, como BAHTTEXT (§4:3.17.7.22), JIS (§4:3.17.7.185) e ASC (§4:3.17.7.11).

Identificadores de idiomas: no WordprocessingML (§4:2.3.2.18) e no DrawingML (§4:5.1.5.3), cada parágrafo e carreira pode ser marcado com um identificador de idioma, permitindo que o aplicativo selecione as ferramentas apropriadas de verificação gramatical e outras funcionalidades específicas ao idioma. Além de um identificador para cada idioma, o OpenXML permite a definição de nomes para um conjunto de caracteres, família de fontes e valor PANOSE para ajudar o aplicativo a escolher um conjunto substitutivo de caracteres adequado, quando não houver compatibilidade local.

4.3 Facilidade de Adoção pelo Desenvolvedor

Um desenvolvedor experiente pode começar a escrever aplicativos simples em OpenXML após algumas horas de leitura da especificação.

Embora a especificação descreva um vasto conjunto de recursos, um aplicativo em OpenXML não precisa ser compatível com todos. A declaração de conformidade (§1:2) requer apenas que um consumidor compatível “não rejeite nenhum documento compatível do tipo de documento [esperado]” e que um produtor compatível “seja capaz de produzir documentos compatíveis” (§1:2.5). Também são fornecidas instruções para a interoperabilidade que explicitam o papel da semântica dos elementos (§1:2.6).

Um aplicativo compatível pode ter uma funcionalidade extremamente específica. Por exemplo, poderia ser um processador de lotes que apenas atualize avisos de direitos autorais em uma coleção de documentos de processador de texto ou um conversor de texto para fala, que compreenda suficientemente uma apresentação de slides e converta o conteúdo de texto em áudio à medida que o usuário passe de um slide ao outro. A estrutura do formato do arquivo permite que programas desse tipo sejam escritos com um mínimo de conhecimentos de OpenXML. Especificamente:

- O formato de arquivo é compatível com padrão largamente adotados, especialmente XML e ZIP, para os quais há ferramentas estabelecidas;
- O formato de arquivo utiliza as convenções para pacotes abertos (Open Packaging Conventions), que combinam XML e ZIP com mecanismos-padrão para expressar os relacionamentos com um arquivo. Em razão disso, é comum poder navegar pelo conteúdo de um arquivo sem conhecer a semântica das marcas de linguagens de marcação primárias ou de apoio em OpenXML;
- Os elementos em níveis profundos da árvore do XML podem ser acessados e modificados sem alterar o resto da estrutura.

Pequenos detalhes por todo o formato de arquivo, alguns deles inexistentes nos formatos binários, são compatíveis com aplicativos de funcionalidade mínima por meio do fornecimento de valores em cache. Por exemplo:

- Sem implementar um paginador, um aplicativo, como um leitor para deficientes visuais, poderia oferecer a navegação em páginas que utilizassem as últimas quebras de página calculadas (§4:2.3.3.13);
- Sem a implementação de fórmulas e a integração com uma fonte externa de dados, um programa de planilhas poderia trabalhar com cálculos em cache (§3:3.2.9) e dados externos em cache (§4:3.14 e §4:3.10.1.76).

Um documento mínimo compatível é muito simples; veja a subseção “Documento mínimo em WordprocessingML”.

4.4 Compacidade

O formato de arquivo OpenXML permite a criação de aplicativos de alto desempenho. Nesta subseção, são descritos alguns pontos do projeto que resultam em um arquivo compacto, acelerando, assim, o processamento e a análise. Na próxima subseção, demonstra-se como a estrutura modular do arquivo permite que um aplicativo realize muitas tarefas pela análise ou modificação de apenas um subconjunto do documento.

Os arquivos OpenXML são tipicamente armazenados em um arquivo ZIP para fins de empacotamento e compactação, de acordo com a implementação recomendada das convenções para pacotes abertos. Surpreendentemente, talvez, os arquivos OpenXML são em média 25% menores, às vezes até 75% menores, que os seus correspondentes binários. Por exemplo, este documento é 85% maior no formato binário!

Uma segunda fonte de compacidade simples, especialmente quando uma representação descompactada for necessária, é o comprimento dos identificadores no XML. Os nomes de marcas usados frequentemente são curtos. Os implementadores são incentivados a usar prefixos curtos de namespace também. Por exemplo, o prefixo convencional para o namespace do WordprocessingML é "w".

Ainda mais compacidade é obtida evitando-se repetições em todo o formato do arquivo. Classes de exemplos eliminam o armazenamento redundante de objetos grandes.

- No SpreadsheetML, seqüências repetidas são armazenadas e indexadas em uma tabela de seqüências na pasta de trabalho (§3:3.3).
- No SpreadsheetML, uma fórmula preenchida em uma ou diversas células é armazenada como uma única fórmula "mestre" na célula superior esquerda. Outras células no intervalo a ser preenchido utilizam-na por um índice de agrupamento (§3:3.2.9.2).
- No DrawingML, os nomes de formas (§4:5.1.12.56), recursos geométricos de texto (§4:5.1.12.76) e outras preconfigurações (diversas em §3:5.8, §3:5.9 e §4:5.1.12) são representadas por nome ou número, em vez de imagem. Nesses casos, os significados dos nomes e números residem na especificação, não no arquivo. Aqui, a representação escolhida é resultado de uma decisão negociada durante o processo de definição de padrões. É compacta e permite a edição no nível correto de abstração: por exemplo, um retângulo poderia ser alterado para uma elipse pela modificação de um único atributo (§4:5.1.11.18).





Em outra classe de exemplos, a hierarquia é usada para fornecer a semântica por herança. Como produto colateral vantajoso, o desempenho é aumentado pela redução do tamanho dos arquivos.

- No WordprocessingML, os estilos são hierárquicos (§3:2.8.9).
- No DrawingML, as formas são agrupadas hierarquicamente (§4:5.1.2.1.20).
- No PresentationML, uma hierarquia-padrão relaciona slides mestres, layouts de slide e slides (§3:4.2).

Outros aspectos do OpenXML também são projetados para proporcionar uma implementação eficiente. Por exemplo, no SpreadsheetML, a tabela de células armazena apenas células com conteúdo e é capaz de representar células mescladas como uma unidade. A economia propiciada por essa técnica é significativa para analisar planilhas.

4.5 Modularidade

Um aplicativo pode realizar muitas tarefas pela análise e modificação de um pequeno subconjunto do documento. Três recursos do formato OpenXML ajudam a proporcionar essa modularidade:

- Os documentos não são blocos monolíticos, mas sim formados por diversas partes.
- Os relacionamentos entre as partes, por sua vez, também são armazenados em partes.
- O formato de arquivo ZIP normalmente usado como apoio aos documentos OpenXML propicia o acesso aleatório a cada parte.

Por exemplo:

- Um aplicativo pode mover um slide (e seus recursos como imagens e layouts) diretamente entre duas apresentações, sem ter de fazer a análise do conteúdo (§3:13.3.8). Esse exemplo implica o uso de dados que acionam relacionamentos explícitos para localizar o slide e os seus recursos. Os relacionamentos explícitos são definidos pelas convenções para pacotes abertos e podem ser analisados sem nenhum conhecimento da semântica das marcas do PresentationML (§1:9.2, §2:8.3).
 - Um aplicativo pode remover todos os comentários de um documento WordprocessingML sem analisar nada dos seus conteúdos (§1:11.3.2). Esse exemplo implica o uso de dados que acionam relacionamentos explícitos para localizar os comentários. Os relacionamentos implícitos são específicos ao OpenXML e portanto não demandam o conhecimento da linguagem de marcação relevante (§1:9.2).
-

4.6 Migração de Alta Fidelidade

O OpenXML é projetado para ser compatível com todos os recursos dos Microsoft Offices 97 a 2003 em formato binário.

O tamanho da dificuldade de se alcançar essa meta é proporcional ao sucesso do resultado: a exclusividade do OpenXML. Alguns formatos, como o PDF, são projetados para produzir uma cópia visualmente exata de um documento para o usuário final. Em vez disso, o objetivo do OpenXML é permitir a edição ou a manipulação no mesmo nível de abstração disponível ao criador original. Por exemplo, reduzir um gráfico vetorial para bitmap não realizaria esse intento, pois desarticulária a hierarquia de estilos em estilos independentes. Ademais, um documento pode obter a semântica computacional que o criador original pretende preservar, como lógicas de fórmulas que dependem de resultados de cálculos intermediários, incluindo códigos de erro e regras de animação que produzem comportamentos dinâmicos.

Essas referências à especificação exemplificam a capacidade do OpenXML de representar aspectos sutis de formatos binários.

- A descrição do SpreadsheetML inclui uma especificação de fórmula extensiva (§4:3.17.7).
- A especificação do WordprocessingML documenta as regras que definem a composição das propriedades de parágrafo, caractere, numeração e tabela na forma de formatação direta (§3:2.8, em particular §3:2.8.10).
- A especificação do PresentationML documenta os recursos de animação (§3:4.4).

O OpenXML permite várias implementações para fins de compatibilidade sem ter de corresponder a cada detalhe irrelevante. Isso é especialmente importante em situações que demandam cálculos numéricos, como layout, processamento de efeitos e avaliação de fórmulas.

A demanda por uma uniformização que exceda a necessidade prática cria obstáculos desnecessários para que os desenvolvedores obtenham a compatibilidade. Essas afirmações são amostras que realçam as decisões tomadas pelo comitê.

- O OpenXML define efeitos, como aparências de superfície (§5.1.12.50), sem obrigar o desenvolvedor a fazer a correspondência desses efeitos pixel por pixel.
- O OpenXML define parâmetros como margens de página (§4:2.6.11), fonte (§4:2.8) e justificação (§4:2.3.1.13). Também permite que os desenvolvedores implementem algoritmos de fluxo diferentes contanto que respeitem esses parâmetros.

- A especificação de fórmulas do SpreadsheetML (§4:3.17.7) não tenta remover as variações de cálculos de ponto flutuante porque, em geral, fazer isso requer que os aplicativos compatíveis implementem emulações lentas em vez de depender do hardware nativo. Ao contrário, um número mínimo de bits de precisão para cálculos numéricos é especificado (§4:3.17.5).
- A especificação de fórmulas do SpreadsheetML também deixa certas decisões condicionais a serem definidas pela implementação, para permitir inovações futuras. Por exemplo, não há limitações quanto ao número de vezes que um cálculo como NORMINV (§4:3.17.7.227) deva ser iterado. (A função NORMINV executa a inversão da distribuição normal pela execução de uma pesquisa iterativa.)

Diversos recursos antigos, como o VML (§3:6), estão incluídos principalmente para proporcionar compatibilidade com versões anteriores. Convém utilizar os padrões mais novos presentes no OpenXML, como o DrawingML (§3:5), para a elaboração de novos documentos.

4.7 Integração com Dados Empresariais

O OpenXML permite que as organizações integrem os aplicativos de produtividade com sistemas de informação que gerenciam processos empresariais, disponibilizando a utilização de esquemas personalizados nos documentos OpenXML. As finalidades de uma organização que adote essa abordagem seriam reutilizar e automatizar o processamento de informações empresariais que, de outra forma, permaneceriam embutidas nos documentos, em locais inacessíveis aos aplicativos, para leitura ou edição. Os aplicativos incluem:

- **Pesquisa:** um usuário final pode pesquisar por empresas com margens de lucro superiores a 20% em uma coleção de planilhas.
- **Marcação de metadados:** uma empresa pode marcar apresentações que foram aprovadas em termos normativos.
- **Montagem de documentos:** um grupo de propostas poderia otimizar a geração desses documentos pela automatização da preparação dos dados subjacentes.
- **Reutilização de dados:** um executivo de vendas pode gerar um relatório de todos os contratos de venda por intervalo de datas listagem de clientes, volume de negócios e quaisquer termos e condições modificados.

- **Aplicativos de linhas de negócios:** os profissionais em um ramo especializado poderiam preparar produtos em um ambiente autoral familiar e ao mesmo tempo escoá-los automaticamente aos sistemas empresariais.

Atingir essas metas requer definir a estrutura e os tipos de dados que uma classe de documentos pode conter e permitir que as informações sejam reveladas sempre que ocorrerem naturalmente no fluxo de cada documento. Considere o exemplo simples de um currículo. Uma estrutura de dados seria definida para que incluísse os campos de nome, telefone, endereço, metas de carreira e qualificações. Além disso, esses campos seriam programados para aparecerem sempre que o usuário os colocasse no documento. Em um outro cenário de negócios, como um grupo financeiro ou centro médico, a estrutura e os campos de dados seriam diferentes.

O OpenXML permite que esse processo ocorra de forma padronizada.

Primeiro, a estrutura dos dados empresariais é expressa utilizando-se um esquema XML padronizado. Isso permite que uma organização expresse os dados com marcas que sejam significativas da perspectiva empresarial. Uma organização pode criar os próprios esquemas ou usar padrões do setor, como o XBRL, para relatórios financeiros, (7) e o HL7, para informações médicas (8). Esquemas estão sendo criados no setor público, dentro de corporações e como padrões do setor, para finalidades que variam desde certidões de nascimento até informações de seguro. Qualquer esquema personalizado pode ser utilizado, contanto que seja expresso na forma de XSD (2).

Segundo, os dados personalizados são embutidos em qualquer documento OpenXML em uma parte Custom XML (§3.7.3), e podem ser descritos em uma parte de propriedades de dados Custom XML (§4:7.5). Ao separar esses dados personalizados da apresentação, o OpenXML permite a um só tempo integrá-los de forma limpa e possibilitar ao usuário final apresentá-los e manipulá-los em uma ampla gama de contextos, como documentos, formulários, slides e planilhas. A interoperabilidade pode, assim, ser obtida em um nível mais fundamental e semanticamente preciso.

4.8 Espaço para Inovação

O OpenXML é projetado para incentivar os desenvolvedores a criar novos aplicativos não implementados na época em que os formatos binários foram definidos ou até mesmo quando o OpenXML foi definido.

Primeiro, discutimos mecanismos de extensibilidade que trabalham em conjunto para permitir a interoperabilidade entre aplicativos com diferentes conjuntos de recursos. Considere um aplicativo de nível superior (com um novo recurso não documentado no OpenXML) e um de nível inferior (que não entenda esse recurso). As três metas primárias da extensibilidade são:

- **Fidelidade visual:** capacidade de o aplicativo de nível inferior exibir o que o de nível superior exibiria. Por natureza, isso requer que um arquivo armazene várias representações do mesmo dado.
- **Possibilidade de edição:** capacidade de editar uma ou mais representações.
- **Privacidade:** capacidade de garantir que as versões anteriores de uma representação não sejam preservadas após a edição de outra representação, que de forma inesperada deixe informações que o usuário achava que haviam sido excluídas ou alteradas. Um aplicativo pode conseguir isso pela eliminação ou a sincronização das representações.

Um desenvolvedor que queira estender o conjunto de recursos do OpenXML tem duas opções principais:

- **Blocos de conteúdo alternativos:** um bloco de conteúdo alternativo (§3:2.18.4 e §5:9.2) armazena várias representações do mesmo conteúdo, cada qual com o seu bloco selecionado. Um aplicativo de nível inferior lê um bloco selecionado de que seja capaz de ler. Durante a edição, ele grava o máximo de blocos selecionados de que é capaz de gravar.
- **Listas de extensões:** a lista de extensões (§3:2.6) armazena o XML personalizado arbitrário sem a representação visual.

Os desenvolvedores têm espaço para inovar além desses mecanismos de extensibilidade.

- **Paradigmas de interação alternativa:** o OpenXML especifica mais que a sintaxe de um documento, mas menos que o comportamento de um aplicativo . Segundo está descrito na declaração de conformidade, o OpenXML enfoca a semântica (§1:2.2, §1.2.3). Logo, um aplicativo compatível fica livre para se comunicar com um usuário final de várias maneiras ou então simplesmente não se comunicar, contanto que a semântica específica seja respeitada.
- **Novos ambientes de processamento:** a declaração de conformidade admite aplicativos de baixa capacidade, para que possam ser executados em dispositivos pequenos, e aplicativos que implementam somente um subconjunto do OpenXML (§1:2.6). O mecanismo de características adicionais permite a produção de aplicativos que informem seus limites de capacidade (§3:8.1).

Como foi indicado na subseção anterior, algumas das oportunidades mais substanciais para a inovação não envolvem o processamento de documentos para interação direta com usuários. Em vez disso, englobam o processamento de máquina a máquina, utilizando formatos de mensagem XML, por exemplo, via XML Web Services (9). Embora não apresentem um comportamento visível ao usuário além das operações com os dados contidos nos documentos OpenXML, esses aplicativos estão sujeitos à compatibilidade do documento (§1:2.4) e à compatibilidade do aplicativo (§1:2.5), que são puramente sintáticas, e às instruções para interoperabilidade (§1:2.6), que incorporam a semântica.

Embora seja impossível listar todas as situações para um processamento de XML personalizado, pode-se antecipar as seguintes: serviços centrados em XML que processem documentos OpenXML para a extração e a inserção automáticas de dados personalizados; serviços de segurança personalizados como assinatura digital XML (10) ou criptografia XML (11); ou mesmo transformações arbitrárias de XSLT (12), ou seja, conversões de/para outros formatos XML. O OpenXML não impõe limitações ou proibições nesse tipo de processamento.

Um dos principais objetivos deste artigo é permitir que o leitor compreenda a estrutura de alto nível de um arquivo OpenXML. Para tal, fornecemos um nível moderado de detalhes sobre as convenções para pacotes abertos (OPC) e menos detalhes sobre as linguagens de marcação individuais.





5. Estrutura de um Documento do Office XML Aberto

As convenções para pacotes abertos (OPC) proporcionam maneiras de armazenar vários tipos de conteúdo (por exemplo, XML, imagens e metadados) em um contêiner, como arquivos ZIP, para representar totalmente um documento. Elas descrevem modelos lógicos para representar conteúdos e relacionamentos.

5.1 Convenções para Pacotes Abertos

A implementação recomendada para as OPC utiliza o formato de arquivo ZIP. Pode-se examinar a estrutura de qualquer arquivo OpenXML através de um visualizador ZIP. Convém examinar o conteúdo de um arquivo OpenXML pequeno por esse método durante a leitura desta descrição. No sistema operacional Windows, basta adicionar a extensão “.zip” ao nome do arquivo e clicar duas vezes nele.

Em termos lógicos, um documento OpenXML é um pacote (§5:8) de OPC. O pacote é uma coleção simples de partes (§5:8.1). Cada parte tem um nome de parte, com diferenciação entre maiúsculas e minúsculas, que consiste em uma seqüência dividida por barras, como “/pres/slides/slide1.xml” (§5:8.1.1). Cada parte também tem um tipo de conteúdo (§5:8.1.2). Em termos físicos, o arquivo ZIP é um pacote, cada item ZIP no arquivo é uma parte e os caminhos e nomes dentro do arquivo ZIP correspondem diretamente aos nomes das partes.

Na implementação ZIP, “[Content_Types].xml” permite que um consumidor determine o tipo de conteúdo de cada parte do pacote (§2:9.2.6). A sintaxe e a definição dos tipos de mídia observam a seção 3.7 do RFC 2616 (13).

Os pacotes e as partes podem conter relacionamentos explícitos (§1:9.2) com outras partes do pacote, bem como com recursos externos. Todos os relacionamentos explícitos têm um ID (para que o conteúdo da parte possa utilizá-lo para fazer referência ao relacionamento) e um tipo (para que o aplicativo possa decidir como processá-lo). Os tipos de relacionamento são nomeados com URIs. Assim, partes não-coordenadas podem criar novos tipos com segurança, sem conflitos.

O conjunto de relacionamentos explícitos de um determinado pacote ou parte de origem é armazenado em uma parte de relacionamento. A parte de relacionamento do pacote como um todo é chamada de “/_rels/.rels”; a parte de relacionamento de uma parte chamada de “/a/b/c.xml” é chamada de “/a/b/_rels/c.xml.rels”. As partes de relacionamento (e, na implementação ZIP, a parte de tipo de conteúdo) são apenas partes com nomes especiais em um pacote. Para abrir um pacote, um aplicativo deve analisar a parte de relacionamento do pacote e seguir os relacionamentos do tipo apropriado.

Todas as outras partes em um documento OpenXML contêm OpenXML, XML personalizado ou conteúdo de tipo arbitrário, como objetos de multimídia. A capacidade da parte de conter XML personalizado é um mecanismo particularmente poderoso para a incorporação de dados empresariais e metadados.

5.2 WordprocessingML

Um documento WordprocessingML é composto de uma coleção de planos (§3:2.1). Os planos são: documento principal (§3:2.2), documento do glossário (§3:2.13), subdocumento (§3:2.18.2), cabeçalho (§3:2.11.1), rodapé (§3:2.11.2), comentário (§3:2.14.5), quadro, caixa de texto (§3:2.18.1), nota de rodapé (§3:2.12.1) e nota de fim de texto (§3:2.12.2).

O único plano obrigatório é o documento principal. É o alvo do relacionamento do pacote do tipo:

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

Um caminho típico, da raiz à folha, na árvore XML é formado por estes elementos XML (§3:2.2):

- **document** – o elemento raiz do documento principal (§3:2.3).
- **body** – corpo (§3:2.7.1). Pode conter vários parágrafos. Também pode conter propriedades de seção especificadas no elemento sectPr.
- **p** – parágrafo (§3:2.4.1). Pode conter uma ou mais carreiras. Também pode conter propriedades de parágrafo especificadas no elemento pPr, que por sua vez contém propriedades-padrão de carreira (também chamadas de propriedades de caractere) especificadas no elemento rPr (§3:2.4.4).
- **r** – carreira (§3:2.4.2). Pode conter vários tipos de conteúdos de carreira, principalmente intervalos de texto. Também pode conter propriedades de carreira (rPr). A carreira é um conceito fundamental no OpenXML, um segmento de texto contíguo com propriedades idênticas; uma carreira não pode conter outras marcações de texto. Por exemplo, uma sentença formada pelas palavras “aqui há três carreiras” seria representada por pelo menos três carreiras: “aqui há”, “três” e “carreiras”. Quanto a isso, o OpenXML difere significativamente dos formatos que permitem o aninhamento arbitrário de propriedades, como o HTML.
- **t** – intervalo de texto (§3:2.4.3.1). Contém uma quantidade arbitrária de texto sem formatação, quebras de linha, tabelas, gráficos e outros elementos não textuais. A formatação do texto é herdada das propriedades da carreira e das propriedades do parágrafo. Este elemento freqüentemente utiliza o atributo `xml:space="preserve"`.

Nesta subseção, discutimos brevemente a formatação direta de texto pela especificação de propriedades de parágrafos e carreiras. A formatação direta enquadra-se no final de um pedido de aplicativo, que também inclui estilos de caracteres, parágrafos, numeração e tabela, bem como padrões de documento (§3:2.8.10). Os próprios estilos são organizados em hierarquias herdadas (§3:2.8.9).

A subseção “Documento WordprocessingML mínimo”, mais à frente, descreve o documento WordprocessingML por inteiro.

5.3 PresentationML

Um documento PresentationML é descrito por uma parte de apresentação. A parte de apresentação é o alvo do relacionamento do pacote do tipo:

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

A apresentação faz referência a esses modelos primários (§3:4.2), que listamos de cima para baixo na hierarquia-padrão:

- Slides mestres, anotações mestras e folhetos mestres (§3:4.2.2), todos herdam propriedades da apresentação;
- Layouts de slide (§3:4.2.5), que herdam propriedades do slide mestre;
- Slides (§3:4.2.3) e anotações (§3:4.2.4), que herdam propriedades dos layouts de slide e anotações mestras, respectivamente.

Cada mestre, layout e slide é armazenado na sua própria parte. O nome de cada parte é especificado na parte de relacionamento da parte de apresentação. Todas as outras seis partes, além da de apresentação, têm essencialmente a mesma estrutura.

Um caminho típico, da raiz à folha, na árvore XML é formado por estes elementos XML (§3:2.2):

- **sld, sldLayout, sldMaster, notes, notesMaster ou handoutMaster** – o elemento raiz.
- **cSld** – slide (§4:4.4.1.15). Pode conter elementos do DrawingML (conforme a descrição nos próximos dois itens) e outros elementos estruturais (conforme descrição a seguir).
- **spTree** – árvore de formas (§4:4.4.1.42). Pode conter propriedades de formas de grupo em um elemento grpSpPr (§4:4.4.1.20) e propriedades de formas de grupo não visuais em um elemento nvGrpSpPr (§4:4.4.1.28). Este nó e todos os seus descendentes são elementos DrawingML. Listamos alguns elementos do DrawingML aqui em razão de sua função central no PresentationML.
- **sp** – forma (§4:4.4.1.40). Pode conter propriedades de formas em um elemento spPr (§4:4.4.1.41) e propriedades de formas não visuais em um elemento nvSpPr (§4:4.4.1.31).

Além do conteúdo da forma do DrawingML, um cSld pode conter outros elementos estruturais, dependendo do elemento raiz no qual reside, como resume a seguinte tabela:

	Slide	Layout de slide	Slide mestre	Folheto mestre	Anotações mestras	Anotações
Dados comuns	X	X	X	X	X	X
Transição	X	X	X			
Horários	X	X	X			
Cabeçalhos e rodapés		X	X	X	X	
Nome correspondente		X				
Tipo de layout		X				
Preservação		X	X			
Lista de layout			X			
Estilo do texto			X			

As propriedades especificadas por objetos em níveis inferiores da hierarquia-padrão (slide mestre, layout de slide, slide) prevalecem sobre as propriedades correspondentes especificadas por objetos de níveis superiores.

Por exemplo, se uma transição não for especificada para um slide, a do layout de slide é aproveitada; se não houver uma especificada no layout, a do slide mestre é usada.

5.4 SpreadsheetML

Um documento SpreadsheetML é descrito em nível alto por uma parte de pasta de trabalho. A parte de pasta de trabalho é o alvo do relacionamento do pacote do tipo:

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

A parte de pasta de trabalho armazena informações sobre a pasta de trabalho e sua estrutura, como versão do arquivo, aplicativo de criação e senha a ser modificada. Em termos lógicos, a pasta de trabalho contém uma ou mais folhas (§3:3.2); em termos físicos, cada folha é armazenada em sua própria parte e referenciada da maneira usual a partir da parte de pasta de trabalho. Cada folha pode ser uma planilha, uma folha de gráfico ou folha de caixa de diálogo. Discutiremos somente a planilha, que é o tipo mais comum. Em um objeto de planilha, um caminho típico, da raiz à folha, na árvore XML é formado por estes elementos XML:

- **worksheet** – o elemento raiz em uma planilha (§3:3.2).
- **sheetData** – a tabela de células, que representa todas as células não vazias na planilha (§3:3.2.4).
- **row** – uma linha de células na tabela de células (§3:2.8).
- **c** – uma célula (§3:3.2.9). O atributo r indica a localização da célula usando coordenadas de estilo A1. A célula também pode ter um identificador de estilo (atributo s) e um tipo de dados (atributo t).
- **v e f** – o valor (§3:3.2.9.1) e a fórmula opcional (§3:3.2.9.2) da célula. Se a célula tiver uma fórmula, o valor é o resultado do cálculo mais recente.

As seqüências e as fórmulas ficam armazenadas em tabelas compartilhadas (§3:3.3 e §3:3.2.9.2.1), para evitar o armazenamento redundante e cargas e gravações de velocidade.

5.5 Linguagens de Marcação Compatíveis

Várias linguagens de marcação compatíveis também podem ser usadas para descrever o conteúdo de um documento OpenXML.

- **DrawingML** (§3:5) – utilizado para representar formas e outros objetos graficamente processados em um documento.
- **VML** (§3:6) – formato para gráficos vetoriais incluído para a compatibilidade com versões anteriores e, por fim, a substituição pelo DrawingML.
- **MLs compartilhados**: matemática (§3:7.1), metadados (§3:7.2), XML personalizado (§3:7.3) e bibliografia (§3:7.4).



5.6 Documento WordprocessingML Mínimo

Esta subseção contém um documento WordprocessingML mínimo, composto de três partes.

A parte de tipo de conteúdo “[Content_Types].xml” descreve os tipos de conteúdo das duas outras partes necessárias.

```
<Types xmlns="HYPERLINK "http://schemas.openxmlformats.org/package/2006/content-types" http://schemas.openxmlformats.org/package/2006/content-types ">
  <Default Extension="rels"
    ContentType="application/vnd.openxmlformats-package.relationships+xml"/>
  <Default Extension="xml"
    ContentType="application/vnd.openxmlformats-officedocument.wordprocessingml.document.main+xml"/>
</Types>
```

A parte de relacionamento do pacote “/_rels/.rels” descreve o relacionamento entre o pacote e a parte do documento principal.

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

A parte do documento, neste caso “/document.xml”, contém o conteúdo do documento.

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

A especificação fornece documentos mínimos e detalhes adicionais do WordprocessingML (§1:11.2), PresentationML (§1:13.2) e SpreadsheetML (§1:12.2).

6. Resumo

O OpenXML é produto dos esforços intensivos de representantes de diversos setores e instituições públicas, de variadas formações e interesses organizacionais. Abrange o conjunto completo de recursos usados do corpus de documentos existentes, e também as necessidades de internacionalização inerentes a todos os principais grupos de idiomas de todo o mundo. Como resultado do trabalho de padronização realizado pelo Ecma TC45 (1) e contribuições via comentários públicos, o OpenXML possibilita um alto grau de interoperabilidade e independência de plataforma; e a sua documentação tornou-se completa (abrangente material de referência) e acessível (pelas descrições não normativas). Também inclui informações suficientes de produtos de tecnologia auxiliares para o processamento adequado de documentos. As implementações em OpenXML podem ser muito pequenas e oferecer uma funcionalidade específica ou englobar todo o conjunto de recursos. Os mecanismos de extensibilidade embutidos no formato garantem que haja espaço para inovações.

Com o tempo, a padronização da especificação e da manutenção do formato garante que vários tipos de usuários possam depender do OpenXML com segurança, e ter certeza de que as evoluções subseqüentes desfrutem a segurança e o equilíbrio resultantes do processo de definição de padrões abertos. Há uma necessidade premente por um padrão aberto de formato de documentos que seja capaz de preservar os bilhões de documentos criados nos formatos binários existentes, bem como os bilhões que continuam sendo criados anualmente. Os avanços tecnológicos nas infra-estruturas de hardware, rede e software baseado em padrões tornam essa missão possível. A explosiva diversificação das demandas do mercado, incluindo os significativos investimentos existentes em sistemas de missão crítica, torna-a essencial.





Para mais informações sobre o Office Open XML, visite:

<http://www.openxmlcommunity.org/community.aspx>

<http://www.openxml4j.com>

<http://openxmldeveloper.org/default.aspx>

<http://www.codeplex.com/NDOS/Wiki/View.aspx>