

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 de processamento de texto, apresentações e planilhas que pode ser implementado gratuitamente por vários aplicativos em diversas plataformas. Sua publicação beneficia as organizações que pretendem implementar aplicativos capazes de usar esse formato, entidades comerciais e governamentais que adquirem tal software e educadores ou autores que ensinam o formato. Assim, todos os usuários desfrutam dos benefícios de um padrão XML para seus documentos, incluindo estabilidade, preservação, interoperabilidade e evolução constante.

O trabalho de padronização do OpenXML foi executado pela Ecma International por meio do TC45 (Comitê Técnico 45), que inclui representantes da Apple, Barclays Capital, BP, The British Library, Essilor, Intel, Microsoft, NextPage, Novell, Statoil, Toshiba e Biblioteca do Congresso dos EUA (1).

Este white paper resume o OpenXML. Leia-o para:

- Entender as finalidades do OpenXML e a estrutura de sua Especificação;
- Conhecer suas propriedades: a forma como ele trata compatibilidade com versões anteriores, preservação, capacidade de extensão, esquemas personalizados, subconfigurações, diversas plataformas, internacionalização e acessibilidade;
- Saber como seguir a estrutura de alto nível de qualquer arquivo OpenXML e navegar rapidamente por qualquer parte da Especificação da qual você necessita mais detalhes.

2 FINALIDADES DO PADRÃO

O OpenXML foi desenvolvido desde o início para ser capaz de representar fielmente os arquivos de documentos de processamento de texto, apresentações e planilhas preexistentes que foram codificados em formatos binários definidos pela Microsoft Corporation. O processo de padronização consistia em espelhamento no XML dos recursos necessários para representar os arquivos já existentes, ampliá-los, fornecer documentação detalhada e habilitar a interoperabilidade. No momento em que um documento estava sendo escrito, mais de 400 milhões de usuários gravavam documentos nos formatos binários, chegando a mais de 40 bilhões de documentos, segundo as estimativas, e mais bilhões a serem criados a cada ano.

Os formatos binários originais para esses arquivos foram criados em uma época em que o espaço era precioso e o tempo de análise afetava severamente a experiência do usuário. Eles baseavam-se em serialização direta de estruturas de dados na memória utilizadas por aplicativos do Microsoft® Office®. As modernas infra-estruturas de padrões (especialmente XML), hardware e rede permitem um novo design que favorece a implementação feita por vários fornecedores em diversas plataformas, levando em consideração a evolução.

Simultaneamente a esses avanços tecnológicos, os mercados têm diversificado para incluir uma nova variedade de aplicativos não contemplados originalmente no mundo simples dos programas de edição de documentos. Esses novos aplicativos incluem aqueles que:

- Geram documentos automaticamente, usando dados corporativos;
- Extraem dados corporativos de documentos e transferem esses dados para aplicativos de negócios;
- Executam tarefas restritas que operam em um subconjunto pequeno de um documento, preservando a capacidade de edição;
- Fornecem acessibilidade para usuários com necessidades especiais, como os deficientes visuais; ou
- São executados em uma variedade de hardwares, incluindo dispositivos móveis.

Talvez a questão mais séria seja a preservação de longo prazo. Aprendemos a criar volumes exponencialmente crescentes de informações. Mesmo assim, estamos codificando essas informações utilizando representações digitais que estão tão profundamente ligadas aos programas que as criaram que, depois de uma ou duas décadas, elas se tornam extremamente difíceis de serem lidas sem uma perda significativa. A preservação de investimentos financeiros e intelectuais em documentos (existentes e novos) tornou-se uma prioridade urgente.

O surgimento destas quatro forças — adoção extremamente ampla de formatos binários, avanços tecnológicos, forças do mercado que exigem vários aplicativos e a crescente dificuldade da preservação de longo prazo — criou uma necessidade de definir um formato XML aberto e de migrar os bilhões de documentos para tal formato, com o mínimo de perda possível. Além disso, a padronização desse formato XML aberto e a sua manutenção ao longo do tempo cria um ambiente, no qual qualquer organização pode confiar seguramente em relação à estabilidade contínua da especificação, confiante de que desenvolvimentos futuros se beneficiarão das limitações e inspeções realizadas por um processo de padrões.

Existem vários padrões e especificações de documentos, os quais incluem HTML, XHTML, PDF e seus subconjuntos, ODF, DocBook, DITA e RTF. Assim como os inúmeros padrões que representam imagens de bitmap, incluindo TIFF/IT, TIFF/EP, JPEG 2000 e PNG, cada um deles foi criado para um conjunto diferente de finalidades. O OpenXML atende à necessidade de um padrão que aborde os recursos representados nos arquivos de documentos já existentes. Até onde sabemos, este é o único formato de documentos XML que oferece suporte a cada um dos recursos dos formatos binários.

O OpenXML define formatos para documentos de processamento de texto, apresentações e planilhas. Cada tipo de documento é especificado por meio de uma linguagem de marcação principal: WordprocessingML, PresentationML ou SpreadsheetML. Mecanismos de incorporação permitem que um documento de qualquer um desses três tipos contenha material nas outras linguagens de marcação principais e em um número de linguagens de marcação com suporte.

A Especificação contém material normativo (que define o OpenXML) e material informativo (que ajuda a compreensão do leitor, mas não é previamente determinado). Ela está estruturada em Partes, a fim de atender às necessidades de públicos diferentes.

Parte 1 — Princípios Básicos 165 páginas	<ul style="list-style-type: none"> — Define vocabulário, convenções de escrita e abreviações. — Resume as três linguagens de marcação principais e as linguagens de marcação com suporte. — Estabelece condições de conformidade e fornece orientações relacionadas a interoperabilidade. — Descreve as limitações das Open Packaging Conventions que se aplicam a cada tipo de documento.
Parte 2 — Open Packaging Conventions 125 páginas	<ul style="list-style-type: none"> — Define as OPC (Open Packaging Conventions). Cada arquivo OpenXML compreende um grupo de fluxos de bytes chamado partes, combinado em um container chamado pacote. O formato dos pacotes é definido pelas OPC. — Descreve uma implementação física recomendada das OPC que utilizam o formato de arquivos ZIP. — Declara os esquemas XML para as OPC como XSD (XML Schema Definitions) (2), em um anexo que é gerado apenas em formato eletrônico. O anexo inclui também representações não-normativas dos esquemas, utilizando RELAX NG (ISO/IEC 19757-2) (3).
Parte 3 — Instruções 466 páginas	<ul style="list-style-type: none"> — Apresenta os recursos de cada linguagem de marcação, fornecendo contexto e ilustrando elementos com exemplos e diagramas. Esta Parte é informativa (não-normativa). — Descreve o recurso de armazenamento de dados XML personalizados em um pacote para oferecer suporte à integração com dados corporativos.
Parte 4 — Referência à linguagem de marcação 5756 páginas	<ul style="list-style-type: none"> — Define cada elemento e atributo, a hierarquia de relações pai/filho entre elementos e semântica adicional, conforme adequado. Esta Parte deve ser utilizada como referência sempre que são necessários detalhes completos sobre um elemento ou atributo.

	<ul style="list-style-type: none"> – Define o recurso de armazenamento de dados XML personalizados. – Declara os esquemas XML para as linguagens de marcação como (XSD) (2), em um anexo que é gerado apenas em formato eletrônico. O anexo também os declara de forma não-normativa, utilizando RELAX NG (ISO/IEC 19757-2) (3).
Parte 5 — Compatibilidade e Capacidade de Extensão das Marcações 34 páginas	<ul style="list-style-type: none"> – Descreve os recursos de extensão de documentos OpenXML. – Especifica os elementos e atributos pelos quais aplicativos com diferentes extensões podem interoperar. – Expressa regras de capacidade de extensão, utilizando NVDL (ISO/IEC 19757-4) (4).

Para facilitar a leitura deste documento e sua navegação, as versões eletrônicas possuem vários vínculos ativos internos. Particularmente, a Parte 4 possui vínculos a elementos pais e filhos.

4 PROPRIEDADES DO PADRÃO

Esta seção prepara o usuário para explorar o OpenXML, descrevendo algumas de suas propriedades de alto nível. Cada subseção descreve uma dessas propriedades e faz referência a recursos específicos no OpenXML.

- “Interoperabilidade”: descreve a forma como o OpenXML independe de formatos proprietários, recursos e ambiente do tempo de execução, permitindo aos desenvolvedores uma ampla gama de opções.
- “Internacionalização”: menciona algumas maneiras significativas nas quais o OpenXML oferece suporte a cada grupo de linguagem principal.
- “Poucas barreiras para adoção feita pelo desenvolvedor”, “Compactação” e “Modularidade”: listam maneiras específicas em quais o OpenXML evita ou remove impedimentos práticos à implementação feita por partes diferentes: curva de aprendizagem, conjunto mínimo de recursos e desempenho.
- “Migração de alta fidelidade”: descreve como o OpenXML cumpre as metas de superarquivamento para preservar as informações, incluindo o objetivo integral do criador original em documentos novos e já existentes.
- “Integração com dados corporativos”: descreve a forma como o Open XML incorpora informações corporativas em esquemas personalizados para habilitar a integração e a reutilização de informações entre aplicativos de produtividade e sistemas de informação.
- “Espaço para inovação”: descreve como o OpenXML prepara o futuro, definindo outros mecanismos de capacidade de extensão e fornecendo interoperabilidade entre aplicativos com conjuntos de recursos diferentes.

O restante deste documento, incluindo esta seção, é um guia típico para o OpenXML. As referências feitas 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 feitas a outros títulos deste documento são feitas por nome.

4.1 INTEROPERABILIDADE

Os desenvolvedores podem escrever aplicativos que consumam e produzam OpenXML em várias plataformas.

Primeiramente, a interoperabilidade do OpenXML foi obtida por meio de extensas contribuições, modificações e revisões da Especificação por parte dos membros do comitê Ecma TC45 (1) com diferentes experiências e interesses corporativos. A Representação incluiu:

- Fornecedores (Apple, Intel, Microsoft, NextPage, Novell e Toshiba) com vários sistemas operacionais (Linux, MacOS e Windows) e várias finalidades de uso para o OpenXML.
- Corporações (BP, Barclays Capital, Essilor, Statoil) com grandes investimentos no conteúdo já existente, incluindo sistemas de transações de missão crítica.
- A British Library e a Biblioteca do Congresso dos Estados Unidos têm interesse em preservação.

Durante a preparação, os membros do comitê levantaram e resolveram centenas de questões relacionadas à política, clareza, semântica e possível dependência do ambiente. Os grupos de questões e outras atividades de representação incluíram:

- Recursos para oferecer suporte a independência de plataformas para mecanismos que eram proprietários nos formatos binários originais.
- Condições para conformidade.
- Conteúdo dos esquemas.
- Representações alternativas para os esquemas e mecanismos de capacidade de extensão, utilizando RELAX NG (ISO/IEC 19757-2) e NVDL (ISO/IEC 19757-4) (4).
- Desenvolvimento de ferramentas de análise automática e visualização de esquemas.
- Internacionalização.
- Integridade, exatidão e clareza das descrições em toda a Especificação, em muitos casos como resultado da tentativa de implementação de partes da Especificação.

O restante desta subseção destaca áreas específicas nas quais o OpenXML deixa os formatos binários originais a favor da interoperabilidade.

Um dos principais requisitos para a interoperabilidade é a independência em relação a qualquer tipo específico de conteúdo de fonte.

- O OpenXML não tem nenhuma restrição em relação a tipos de imagem, áudio ou vídeo. Por exemplo, as imagens podem estar no formato GIF, PNG, TIFF, PICT, JPEG ou qualquer outro tipo de imagem (§1:14.2.12).
- Os controles incorporados podem ser de qualquer tipo, como Java ou ActiveX (§1:15.2.8).
- As especificações de fontes WordprocessingML podem incluir métricas de fontes e informações PANOSE para ajudar a descobrir uma fonte substituta se a original não estiver disponível (§3:2.10.5).

Além disso, o OpenXML evita dependência no ambiente do tempo de execução do aplicativo que produziu um documento.

- O exemplo clássico é um aplicativo de controle externo que gera uma imagem para uma parte da superfície de exibição. Para evitar casos em que o controle ou o aplicativo não esteja disponível ou não possa ser executado em um ambiente do tempo de execução específico, o arquivo de documentos pode conter uma representação de imagem. Esse mecanismo existe também nos formatos binários mais antigos.

- O OpenXML apresenta um mecanismo mais geral, chamado Bloco de conteúdo alternativo (§3:2.18.4), que pode ser utilizado em várias situações em que um aplicativo de consumo pode não ser capaz de interpretar o que foi escrito por um aplicativo de produção. Ele é utilizado principalmente no contexto de capacidade de extensão. Esse mecanismo é descrito na subseção “Espaço para inovação”.
- Além disso, o OpenXML evita dependência de qualquer parâmetro que seja significativo no ambiente do produtor do documento, mas que não está no ambiente do consumidor. Por exemplo, o parâmetro CT_SYSCOLOR é um índice em uma tabela de cores no ambiente de produção. Para oferecer suporte à portabilidade a um ambiente de consumo diferente, o tipo PresentationML permite que o produtor armazene em cache a cor do sistema que estava sendo utilizada no momento em que o documento foi criado.

Finalmente, e mais importante, o Office OpenXML atende aos padrões W3C abertos, como XML (5) e XML Namespaces (6). Esse fato, por si só, permite um nível básico de interoperabilidade em todas as plataformas e sistemas operacionais que aderem a esses padrões abertos.

4.2 INTERNACIONALIZAÇÃO

O OpenXML oferece suporte a recursos de internacionalização exigidos por diversos idiomas, como árabe, chinês (as três variantes), hebraico, hindi, japonês, coreano, russo e turco.

O OpenXML oferece suporte inerente ao Unicode, pois ele é um XML. Além disso, o OpenXML possui um vasto conjunto de recursos de internacionalização que foram redefinidos ao longo de vários anos. A lista é significativa:

Orientação de texto: o OpenXML oferece suporte a idiomas escritos da esquerda para a direita (LTR) e da direita para a esquerda (RTL). Também oferece suporte a idiomas bidirecionais (“BiDi”), como árabe, farsi, urdu, hebreu e yiddish, que são executados da direita para a esquerda, mas que contêm segmentos incorporados de texto executado da esquerda para a direita. Em WordprocessingML, a direção do texto pode ser controlada tanto em nível de parágrafo (§4:2.3.1.6) quanto em nível de série em um parágrafo (§4:2.3.2.28). Da mesma forma, em DrawingML, a direção do texto pode ser controlada em nível de corpo (§4:5.1.5.1.1), em nível de parágrafo (§4:5.1.5.2.2) e em marcadores numerados (§4.5.1.5.4).

Fluxo de texto: em WordprocessingML, a direção do fluxo do texto pode ser controlada em nível de uma seção ou uma tabela (§4:2.3.1.41) ou em nível de um parágrafo (§4:2.3.2.28). Nos níveis de seção e tabela, o fluxo de texto pode ser controlado nas direções vertical e horizontal. Isso permite que o OpenXML ofereça suporte a todos os possíveis layouts de texto (por exemplo linhas verticais fluindo de cima para baixo e empilhadas da esquerda para a direita para oferecer suporte a Mongolian). Isso afeta o layout de listas, tabelas e outros elementos de apresentação. DrawingML também utiliza definições Kumimoji nos níveis de parágrafo e série de modo que o texto flua horizontalmente e os números fluam verticalmente (§4:5.1.5.2.3, §4:5.1.5.3.9). Em WordprocessingML (§4:2.3.1.16) e PresentationML (§4:4.3.1.15), o fluxo de caracteres também pode ser especificado utilizando-se as definições Kinsoku para especificar quais caracteres que podem começar e terminar uma linha de texto.

Representação de números: para formatação de campo em WordprocessingML (§4:2.16.4.3), numeração de parágrafos/lista em WordprocessingML (§4:2.9) e numeração em DrawingML (§4:5.1.5.4, §4:5.1.12.61), os números podem ser formatados utilizando-se qualquer um dos vários formatos de número, incluindo hiragana, árabe, abjad, tailandês, texto cardinal (por exemplo, "cento e vinte e três"), chinês, coreano (chosung ou ganada), hebreu, hindija, japonês, romano ou vietnamita. Esses recursos também oferecem suporte a valores de pontos base arbitrários (por exemplo "1.00" vs. "1,00") e separadores de listas. A formatação internacionalizada de números é particularmente robusta em SpreadsheetML, que oferece suporte a todos esses recursos nos formatos de células (§4:3.8.30) e em referência a dados externos (§4.3.13.12).

Representação de datas: em WordprocessingML (§4:2.18.7) e SpreadsheetML (§4:3.18.5), as datas de calendário podem ser escritas usando formatos gregoriano (três variações), hebreu, hijri, japonês (era do imperador), coreano (era de Tangun), saka, de Twaian e tailandês.

Fórmulas: a especificação de fórmulas em SpreadsheetML fornece várias 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: em WordprocessingML (§4:2.3.2.18) e DrawingML (§4:5.1.5.3), cada parágrafo pode ser marcado com um identificador de idioma, permitindo que um aplicativo selecione ferramentas de revisão adequadas e outras funcionalidades específicas do idioma. Além de um identificador para cada idioma, o OpenXML oferece suporte à nomeação de um conjunto de caracteres, uma família de fontes e um valor PANOSE para auxiliar o aplicativo na escolha de um conjunto de caracteres substituto adequado quando não há suporte local.

4.3 POUCAS BARREIRAS PARA ADOÇÃO FEITA PELO DESENVOLVEDOR

Um desenvolvedor experiente pode começar a escrever aplicativos simples compatíveis com Open XML poucas horas depois de começar a ler a Especificação.

Embora a Especificação descreva um grande conjunto de recursos, um aplicativo em conformidade com o Open XML não necessita do suporte de todos os recursos descritos. A Declaração de conformidade (§1:2) exige apenas que um consumidor em conformidade "não rejeite qualquer documento de conformidade do tipo de documento [esperado]" e que um produtor em conformidade "possa produzir documentos de conformidade" (§1:2.5). Ela também fornece orientações de interoperabilidade que especificam a função da semântica de elementos (§1:2.6).

Um aplicativo compatível pode ter funcionalidades extremamente concentradas. Por exemplo, poderia ser um processador em lote que simplesmente atualize as notas de direitos autorais em um grupo de documentos de processamento de texto ou uma leitora de texto para fala que entenda uma apresentação em slides para processar seu conteúdo de texto em áudio à medida que o usuário navega entre os slides. A estrutura do formato de arquivo permite que tais programas sejam escritos com um mínimo conhecimento do OpenXML. Especificamente:

- O formato de arquivos está em conformidade com padrões bem estabelecidos, especialmente XML e ZIP, para os quais existem ferramentas desenvolvidas.
- O formato de arquivos utiliza Open Packaging Conventions, que combinam XML e ZIP com mecanismos padrão para expressar as relações em um arquivo. Por causa disso, o conteúdo de um arquivo pode, em geral, ser navegado sem conhecimento da semântica de tags para qualquer uma das linguagens de marcação principais e de suporte no OpenXML.
- Os elementos na base da árvore XML podem ser acessados e modificados sem interferir no restante da estrutura.

Pequenos detalhes nos formatos de arquivos, alguns dos quais não estavam presentes nos formatos binários, oferecem suporte a aplicativos com o mínimo de funcionalidade, fornecendo valores em cache. Por exemplo:

- Sem a implementação de um paginador, um aplicativo, como uma leitora para cegos, poderia oferecer navegação entre páginas utilizando as últimas quebras de página calculadas (§4:2.3.3.13).
- Sem a implementação de fórmulas e a integração com uma fonte de dados externos, um programa de planilhas poderia funcionar a partir de cálculos em cache (§3:3.2.9) e de dados externos em cache (§4:3.14 e §4:3.10.1.76).

Um documento compatível mínimo é extremamente simples; consulte a subseção “Documento WordprocessingML mínimo”.

4.4 COMPACTAÇÃO

O formato de arquivos do OpenXML oferece suporte à criação de aplicativos de alto desempenho. Nesta subseção, descrevemos alguns dos pontos de design que resultam em um arquivo compacto, agilizando, assim, a manipulação e a análise. Na próxima subseção, mostramos como a estrutura de arquivo modular permite que um aplicativo realize muitas tarefas analisando ou modificando apenas um pequeno subconjunto de um documento.

Um arquivo do OpenXML é convencionalmente armazenado em um arquivo ZIP para fins de empacotamento e compactação, seguindo a implementação recomendada das Open Packaging Conventions. Talvez com alguma surpresa, os arquivos Open XML são, em média, 25% menores e, às vezes, até 75% menores que os arquivos em formato binário. Por exemplo, este white paper é 85% maior que o formato binário!

Uma segunda fonte simples de compactação, particularmente onde uma representação não compactada é necessária, é a extensão dos identificadores no XML. Os nomes das tags usadas com mais frequência são curtos. Os implementadores são encorajados a utilizarem prefixos de namespace pequenos também; por exemplo, o prefixo convencional para o namespace de WordprocessingML é “w”.

Mais compactação é alcançada evitando-se a repetição em todo o formato do arquivo. Uma classe de exemplos remove o armazenamento redundante de grandes objetos.

- Em SpreadsheetML, cadeias de caracteres repetidas são armazenadas em uma tabela de cadeias de caracteres na planilha e são referenciadas por índice (§3:3.3).
- Em SpreadsheetML, uma fórmula que é assumida em várias células é armazenada como uma fórmula “mestra” simples na célula superior esquerda; as outras células na faixa de preenchimento a consultam por um índice de grupos (§3:3.2.9.2).
- Em DrawingML, nomes de formas (§4:5.1.12.56), geometrias de texto (§4:5.1.12.76) e outras predefinições (várias em §3:5.8, §3:5.9 e §4:5.1.12) são representadas por nome ou número em vez de explicitamente. Nesses casos, os significados de nomes e números residem na Especificação, e não no arquivo. Aqui, a representação escolhida é o resultado de uma decisão comercial explícita durante o processo de padrões. Ela é compacta e permite edição no nível correto de abstração: por exemplo, um retângulo poderia ser alterado para uma oval, alterando-se um único atributo (§4:5.1.11.18).

Em uma outra classe de exemplo, a hierarquia é utilizada para fornecer semântica de herança. Isso aumenta o desempenho, reduzindo o tamanho dos arquivos.

- Em WordprocessingML, os estilos são hierárquicos (§3:2.8.9).
- Em DrawingML, as formas são agrupadas hierarquicamente (§4:5.1.2.1.20).
- Em PresentationML, uma hierarquia padrão faz a relação entre slides mestres, layout de slides e slides (§3:4.2).

Outros aspectos do OpenXML também são desenvolvidos para permitir uma implementação eficiente. Por exemplo, em SpreadsheetML, a tabela de células armazena apenas células que não estejam vazias e é capaz de representar células mescladas como uma unidade. A economia que essa tecnologia possibilita é significativa para planilhas esparsas.

4.5 MODULARIDADE

Um aplicativo pode realizar muitas tarefas analisando ou modificando um pequeno subconjunto de um documento.

Três recursos do formato OpenXML cooperam para fornecer essa modularidade:

- Um documento não é monolítico; ele é criado sem partes múltiplas.
- As relações entre as partes são armazenadas em partes.
- O formato de arquivos ZIP que é geralmente utilizado para oferecer suporte a documentos OpenXML suporta acesso aleatório a cada parte.

Por exemplo:

- Um aplicativo poderia mover um slide de uma apresentação para outra sem qualquer problema, juntamente com recursos como imagens e layouts, sem qualquer análise do conteúdo dos slides (§3:13.3.8). Esse exemplo utiliza dados chamados relações explícitas para localizar o slide e seus recursos. As relações explícitas são definidas pelas Open Packaging Conventions e podem ser analisadas sem qualquer conhecimento da semântica de tags do PresentationML (§1:9.2, §2:8.3).
- Um aplicativo pode remover todos os comentários de um documento WordprocessingML sem analisar seu conteúdo (§1:11.3.2). Esse exemplo utiliza dados chamados relações implícitas para localizar os comentários. As relações implícitas são específicas do OpenXML e, portanto, exigem certo conhecimento da linguagem de marcação relevante (§1:9.2).

4.6 MIGRAÇÃO DE ALTA FIDELIDADE

O OpenXML foi desenvolvido para oferecer suporte a todos os recursos dos formatos binários do Microsoft Office 97-2003.

É difícil exagerar a dificuldade de se atingir essa meta e a conseqüente exclusividade do OpenXML em fazer isso. Alguns formatos, como o PDF, foram desenvolvidos para oferecer a um usuário final uma cópia visual exata de um documento acabado. Por outro lado, o OpenXML deve permitir edição ou manipulação futuras no mesmo nível de abstração disponível para o criador original; por exemplo, a redução de gráfico de vetores para um bitmap não atenderia a essa finalidade, como aconteceria reduzindo-se uma hierarquia de estilos para estilos independentes. Além disso, um documento pode conter semântica computacional que o criador do original espera preservar, como a lógica das fórmulas que depende de resultados de cálculos intermediários, incluindo códigos de erros ou regras de animação que produzem comportamento dinâmico.

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

- A descrição SpreadsheetML inclui uma especificação extensa de fórmulas (§4:3.17.7).
- A especificação de WordprocessingML documenta as regras pelas quais as propriedades de parágrafos, caracteres, numeração e tabelas são compostas por formatação direta (§3:2.8, especialmente §3:2.8.10).
- A especificação de PresentationML documenta os recursos de animação (§3:4.4).

O OpenXML permite várias implementações de conformidade sem ter que corresponder a cada detalhe irrelevante. Isso é particularmente importante quando cálculos numéricos estão envolvidos, como layout, processamento de efeitos e avaliação de fórmulas. Exigir mais consistência do que o praticável criaria uma barreira desnecessariamente alta para os desenvolvedores atingirem conformidade. Estas afirmações ressaltam amostras de decisões tomadas pelo comitê em relação a esse assunto.

- O OpenXML define efeitos, como aparências das superfícies (§5.1.12.50), sem impedir que um desenvolvedor associe esses efeitos pixel por pixel.
- O OpenXML define parâmetros, como margens de páginas (§4:2.6.11), fonte (§4:2.8) e justificação (§4:2.3.1.13). Ele permite que os desenvolvedores implementem diferentes algoritmos de fluxos, desde que esses parâmetros sejam respeitados.
- A especificação das fórmulas de SpreadsheetML (§4:3.17.7) não tenta remover as variações em computação de pontos flutuantes porque, em geral, isso exigiria que os aplicativos compatíveis implementassem emulação lenta em vez de contar com hardware nativo. Em vez disso, ela especifica o número mínimo de bits de precisão para cálculos numéricos (§4:3.17.5).
- A especificação de fórmulas SpreadsheetML também deixa certas decisões condicionais definidas por implementação para permitir inovação futura. Por exemplo, ela não limita o número de vezes que um cálculo, como NORMINV (§4:3.17.7.227), deve interagir. (A função NORMINV executa o inverso da distribuição normal, executando uma pesquisa interativa.)

Vários recursos mais antigos, como VML (§3:6), são incluídos, principalmente para compatibilidade com versões anteriores. O uso de padrões mais novos já no OpenXML, como DrawingML (§3:5), é encorajado durante a criação de novos documentos.

4.7 INTEGRAÇÃO COM DADOS CORPORATIVOS

O Open XML permite que as organizações integrem aplicativos de produtividade com sistemas de informação que gerenciam processos corporativos ao habilitar o uso de esquemas personalizados em documentos OpenXML. Os objetivos de uma organização ao adotar essa abordagem é o de reutilizar e automatizar o processamento de informações comerciais que, de outra forma, ficaria oculto de forma opaca dentro de documentos, nos quais aplicativos de negócios não poderiam ler ou gravar.(1)

Os aplicativos incluem:

- *Pesquisa*: um usuário final poderá pesquisar um conjunto de planilhas para obter as empresas que têm margens de lucro superiores a 20%.
- *Tags de metadados*: uma empresa poderá marcar apresentações que foram aprovadas sob uma perspectiva regulamentar.
- *Conjunto de documentos*: um grupo de propostas poderá melhorar a geração de propostas, automatizando a preparação dos dados básicos.

- *Reutilização de dados*: um executivo de vendas poderá gerar um relatório de todos os contratos de vendas em um determinado período, listando clientes, tamanho dos negócios e qualquer termo e condições modificados.
- *Aplicativos de linhas de negócio*: os profissionais em um ramo especializado poderão preparar entregas em um ambiente de autorias familiar, mantendo o fluxo automático dos produtos de trabalho para os sistemas corporativos.

O cumprimento dessas metas exige a definição da estrutura e do tipo de dados que uma classe de documentos pode conter e que as informações possam ser reveladas sempre que ocorrerem naturalmente no fluxo de cada documento. Considere o exemplo simples de um curriculum vitae. Um usuário definiria uma estrutura de dados que incluísse campos chamados nome, número do telefone, endereço, objetivos profissionais e qualificação. Um usuário, então, organizaria esses campos de modo que eles aparecessem sempre que autores humanos os colocassem em um documento. Em uma configuração comercial diferente, como um grupo financeiro ou um centro médico, a estrutura e os campos de dados seriam diferentes.

O OpenXML permite que esse processo ocorra de modo padronizado.

Primeiramente, a estrutura dos dados comerciais é expressa primeiro utilizando um esquema XML personalizado. Isso permite que uma organização expresse os dados com tags que sejam significativas sob uma perspectiva comercial. Uma organização pode criar os próprios esquemas ou utilizar esquemas de padrões industriais, como XBRL para emissão de relatórios financeiros (7) e HL7 para informações médicas (8). Os esquemas estão sendo criados no setor público, dentro das empresas, e como padrões industriais, para fins que variam desde certidões de nascimento até informações sobre seguro. Qualquer esquema personalizado pode ser utilizado, desde que expresso na forma XSD (2).

Segundo, os dados personalizados são incorporados em qualquer documento OpenXML em uma parte XML personalizadas (§3.7.3) e podem ser descritos utilizando a parte Propriedades de dados XML personalizadas (§4:7.5). Separando esses dados personalizados da apresentação, o OpenXML permite a integração transparente de dados, possibilitando ainda a apresentação de usuários finais e a manipulação em uma ampla variedade de contextos, incluindo documentos, formulários, slides e planilhas. A interoperabilidade pode, assim, ser obtida em um nível mais fundamental e semanticamente mais preciso.

4.8 ESPAÇO PARA INOVAÇÃO

O Open XML foi desenvolvido para estimular os desenvolvedores a criar novos aplicativos que não foram contemplados quando os formatos binários foram definidos, ou mesmo quando o Open XML foi definido.

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

- *Fidelidade visual*: a capacidade do aplicativo de nível inferior de mostrar o que um aplicativo de nível superior exibiria. Isso exige inerentemente que um arquivo armazene várias representações dos mesmos dados.
- *Capacidade de edição*: a capacidade de editar uma ou mais representações.

- *Privacidade*: a capacidade de garantir que versões antigas de uma representação não permaneçam após a edição de uma outra representação, deixando inesperadamente as informações que um usuário acredita terem sido excluídas ou modificadas. Um aplicativo pode conseguir isso eliminando ou sincronizando representações.

Um desenvolvedor desejando ampliar 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 um no bloco escolhido. Um aplicativo de nível inferior lê um bloco escolhido que seja capaz de ler. Após a edição, ele grava quantos blocos escolhidos ele puder gravar.
- *Listas de extensão*: uma lista de extensão (§3:2.6) armazena XML personalizado arbitrário sem uma representação visual.

Os desenvolvedores possuem espaço para inovação fora desses mecanismos de capacidade de extensão.

- *Paradigmas de interação alternativos*. O OpenXML especifica mais que uma sintaxe de documentos, mas menos que o comportamento do aplicativo. Como descrito na Declaração de conformidade, ele se concentra na semântica (§1:2.2, §1.2.3). Conseqüentemente, um aplicativo compatível pode ser comunicar livremente com um usuário final de diversas maneiras ou pode não se comunicar com um usuário final de modo algum — desde que a semântica especificada seja respeitada.
- *Ambientes de computação Novel*. A Declaração de conformidade admite aplicativos que tenham pouca capacidade, de modo que eles possam ser executados em pequenos dispositivos, e aplicativos que implementem apenas um subconjunto do OpenXML (§1:2.6). O mecanismo Características adicionais permite que um aplicativo de produção informe seus limites de capacidade (§3:8.1).

Conforme indicado na subseção anterior, algumas das mais importantes oportunidades para inovação não envolvem processamento de documentos para interação direta do usuário. Em vez disso, envolvem processamento máquina para máquina, usando formatos de mensagens XML, por exemplo, por meio do Web Services XML (9). Embora tais aplicativos não tenham um comportamento visível ao usuário diferente das operações em dados contidos em documentos OpenXML, eles estão sujeitos a conformidade de documentos (§1:2.4) e conformidade de aplicativos (§1:2.5), que são puramente sintáticos, e a orientações de interoperabilidade (§1:2.6), que incorporam semântica.

Embora seja impossível enumerar todos os casos de uso possíveis para processamento personalizado do XML, é possível prever serviços centrais do XML que processam documentos OpenXML para extração automática e inserção de dados personalizados, serviços de segurança personalizados, como XML Digital Signature (10) ou XML Encryption (11), ou até mesmo transformação XSLT arbitrária (12) que fazem a conversão em e de outros formatos XML. O OpenXML não impõe nenhuma proibição ou limitação a esse processamento.

5 ESTRUTURA DE UM DOCUMENTO OFFICE OPEN XML

Um dos principais objetivos deste white paper é permitir que o leitor siga a estrutura de alto nível de qualquer arquivo OpenXML. Para fazer isso, fornecemos um nível moderado de detalhes relacionados às OPCs (Open Packaging Conventions) e menos detalhes em relação às linguagens de marcação individuais.

5.1 OPEN PACKAGING CONVENTIONS

As OPCs (Open Packaging Conventions) fornecem uma forma de armazenar vários tipos de conteúdo (por exemplo, XML, imagens e metadados) em um container, como um arquivo ZIP, para representar inteiramente um documento. Elas descrevem um modelo lógico para representar detenção e relações.

A implementação recomendada das OPC utilizam o formato de arquivos ZIP. É possível verificar a estrutura de qualquer arquivo OpenXML utilizando qualquer visualizador de ZIP. A verificação do conteúdo de um arquivo OpenXML pequeno feita dessa maneira é útil durante a leitura dessa descrição. No sistema operacional Windows, basta acrescentar a extensão “.zip” ao nome do arquivo e clicar duas vezes nele.

Do ponto de vista lógico, um documento OpenXML é um *pacote de OPC* (§5:8). Um pacote é um grupo simples de *partes* (§5:8.1). Cada parte possui um *nome de parte* que não diferencia letras maiúsculas de minúsculas e que consiste em uma seqüência delimitada por barras de nomes de segmentos, como “/pres/slides/slide 1 .xml” (§5:8.1.1). Cada parte também possui um *tipo de conteúdo* (§5:8.1.2). Fisicamente, o arquivo ZIP é um único pacote, cada item do ZIP no arquivo é uma única parte, e os nomes de caminhos no arquivo ZIP correspondem diretamente a nomes de partes.

Na implementação de 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 definição de tipos de mídia estão na seção 3.7 de RFC 2616 (13).

Os pacotes e as partes podem conter *relações explícitas* (§1:9.2) a outras partes no pacote, bem como para recursos externos. Cada relação explícita possui um ID de relação, que permite que o conteúdo de uma parte faça referência a ela, permitindo que um aplicativo decida como processá-la. Os tipos de relações são nomeados utilizando URIs, permitindo que as partes não coordenadas criem novos tipos com segurança e sem conflitos.

O conjunto de relações explícitas de um pacote de origens ou uma parte específica é armazenado em uma *parte de relações*. A parte de relações do pacote como um todo é chamada “/_rels/.rels”; a parte de relações para uma parte chamada “/a/b/c.xml” é chamada “/a/b/_rels/c.xml.rels”. As partes de relações (e na implementação ZIP, a parte do tipo de conteúdo) são as únicas partes especialmente nomeadas em um pacote. Para abrir um pacote, um aplicativo deve analisar a parte de relações do pacote e seguir as relações de tipo adequado.

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

5.2 WORDPROCESSINGML

Um documento WordprocessingML é composto de um grupo de histórias (§3:2.1). Cada história é uma dos seguintes itens: o documento principal (§3:2.2), o documento de glossário (§3:2.13), um subdocumento (§3:2.18.2), um cabeçalho (§3:2.11.1), um rodapé (§3:2.11.2), um comentário (§3:2.14.5), uma moldura, uma caixa de texto (§3:2.18.1), uma nota de rodapé (§3:2.12.1) ou uma observação final (§3:2.12.2).

A única história exigida é o documento principal. É o alvo da relação de pacotes, cujo tipo é:

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

Um caminho típico desde a raiz até a folha na árvore XML abrangeria 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. Pode conter também propriedades de seção especificadas em um elemento `sectPr`.
- `p` — parágrafo (§3:2.4.1). Pode conter uma ou mais séries. Podem conter também propriedades de parágrafo especificadas em um elemento `pPr`, que, por sua vez, pode conter propriedades de séries padrão (também conhecidas como propriedades de caracteres) especificadas em um elemento `rPr` (§3:2.4.4).
- `r` — série (§3:2.4.2). Pode conter vários tipos de conteúdo de séries, principalmente faixas de texto. Pode também conter propriedades de séries (`rPr`). A série é um conceito básico no OpenXML. Uma série é uma parte contígua de texto com propriedades idênticas; uma série não contém nenhuma marcação de texto adicional. Por exemplo, se uma sentença tivesse que conter as palavras “essa é uma série de **três**”, ela seria representada por pelo menos três séries: “esta é”, “**três**” e “série”. Em relação a isso, o OpenXML se diferencia significativamente de formatos que permitem o aninhamento arbitrário de propriedades, como HTML.
- `t` — faixa de texto (§3:2.4.3.1). Contém um volume arbitrário de texto sem formatação, quebras de linhas, tabelas, gráficos ou outro material que não seja texto. A formatação do texto é herdada das propriedades de série e das propriedades de parágrafos. Esse elemento geralmente utiliza o atributo `xml:space="preserve"`.

Nesta subseção, tratamos da formatação direta de texto, especificando propriedades de parágrafo e séries. A formatação direta fica no final de uma ordem de aplicativo que também inclui estilos de parágrafos, caracteres, numeração e tabelas, bem como padrões de documentos (§3:2.8.10). Esses estilos são organizados em hierarquias de herdadas (§3:2.8.9).

A subseção “Documento WordprocessingML mínimo” a seguir lista um documento WordprocessingML por inteiro.

5.3 PRESENTATIONML

Um documento PresentationML é descrito por uma parte de apresentação. A parte de apresentação é o alvo da relação de pacotes, cujo tipo é:

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

A apresentação refere-se a estes construtores primários (§3:4.2), que listamos de cima para baixo na hierarquia padrão:

- Slides mestres, anotações mestras e apostilas mestras (§3:4.2.2), todas elas herdando propriedades da apresentação;
- Layout de slides (§3:4.2.5), que herdam propriedades do slide mestre e;
- Slides (§3:4.2.3) e páginas de anotações (§3:4.2.4), que herdam propriedades dos layouts mestres e das anotações mestras, respectivamente.

Cada mestre, layout e slide é armazenado na própria parte. O nome de cada parte é especificada na parte da relação da parte da apresentação. Cada uma das seis partes que não seja da apresentação é estruturada basicamente da mesma maneira. Um caminho típico desde a raiz até a folha na árvore XML abrangeria estes elementos XML (§3:2.2):

- `sld`, `sldLayout`, `sldMaster`, `notes`, `notesMaster`, or `handoutMaster` — o elemento raiz.
- `cSld` — slide (§4:4.4.1.15). Pode conter elementos `DrawingML` (conforme descrito nos dois próximos itens) e outros elementos estruturais (como descrito a seguir).
- `spTree` — árvore de formas (§4:4.4.1.42). Pode conter propriedades de formas de grupos em um elemento `grpSpPr` (§4:4.4.1.20) e propriedades de formas de grupos não-visuais em um elemento `nvGrpSpPr` (§4:4.4.1.28). Esse nó e seus descendentes são todos elementos `DrawingML`. Listamos aqui alguns elementos `DrawingML`, devido à sua função básica em `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 de formas `DrawingML`, um elemento `cSld` pode conter outros elementos estruturais, dependendo do elemento raiz em que ele reside, conforme resumido nesta tabela:

	Slide	Layout do Slide	Slide Mestre	Apostila Mestre	Anotações Mestras	Páginas de Anotações
Dados Comuns	X	X	X	X	X	X
Transição	X	X	X			
Tempo	X	X	X			
Cabeçalhos e Rodapés		X	X	X	X	
Nome Correspondente		X				
Tipo de Layout		X				
Preservação		X	X			
Lista de Layouts			X			
Estilo de Texto			X			

As propriedades especificadas por objetos inferiores na hierarquia padrão (slide mestre, layout do slide, slide) substituem as propriedades correspondentes especificadas por objetos em um nível superior na hierarquia. Por exemplo, se uma transição não está especificada para um slide, ela é retirada do layout do slide; se ela não está especificada lá, é retirada do slide mestre.

5.4 SPREADSHEETML

Um documento `SpreadsheetML` é descrito no nível superior por uma parte de planilha. A parte da planilha é o alvo da relação de pacotes, cujo tipo é:

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

A parte de planilha armazena informações sobre a planilha e sua estrutura, como versão do arquivo, criação de aplicativo e senha a ser modificada. Do ponto de vista lógico, a planilha contém uma ou mais folhas (§3:3.2); fisicamente, cada folha é armazenada na própria parte e é consultada normalmente na parte da planilha. Cada folha pode ser uma planilha, uma folha de gráficos ou uma folha de diálogos. Discutiremos apenas a planilha, que é o tipo mais comum. Em um objeto de planilha, um caminho típico desde a raiz até a folha na árvore XML abrangeria estes elementos XML:

- `worksheet` — o elemento raiz em uma planilha (§3:3.2).
- `sheet Data` — a tabela de células, que representa cada célula não-vazia da 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, utilizando coordenadas de estilo A1. A célula pode também ter um identificador de estilos (atributos) 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 uma célula possui uma fórmula, o valor é o resultado do cálculo mais recente.

Tanto a cadeia de caracteres quanto as fórmulas são armazenadas em tabelas compartilhadas (§3:3.3 e §3:3.2.9.2.1) para evitar armazenamento redundante e para agilizar os carregamentos e os salvamentos.

5.5 LINGUAGENS DE MARCAÇÃO DE SUPORTE

Várias linguagens de marcação de suporte também podem ser utilizadas para descrever o conteúdo de um documento OpenXML.

- DrawingML (§3:5) — utilizada para representar formas e outros objetos processados graficamente em um documento.
- VML (§3:6) — formato para gráficos de vetores incluído para fins de compatibilidade com versões anteriores e que poderá ser substituído por DrawingML.
- MLs compartilhadas: 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 que abrange três partes.

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

```
<Types xmlns="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 relações de pacotes “/_rels/.rels” descreve a relação entre o pacote e a parte principal do documento.

```
<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>
```


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 relativos a WordprocessingML (§1:11.2), PresentationML (§1:13.2), and SpreadsheetML (§1:12.2).

6 RESUMO

O OpenXML é o produto do esforço substancial dos representantes de vários setores e instituições públicas com deferentes conhecimentos e interesses organizacionais. Ele cobre todo o conjunto de recursos utilizados no arquivo de documentos já existente, bem como as necessidades de internacionalização inerentes em todos os principais grupos de idiomas do mundo inteiro. Como resultado do trabalho de padronização desempenhado pelo Ecma TC45 (1) e das contribuições recebidas de comentários públicos, o OpenXML tem possibilidade um alto nível de interoperabilidade e independência de plataformas; sua documentação tornou-se completa (por meio de um extenso material de referência) e acessível (por meio de descrições não-normativas). Inclui também informações suficientes para produtos de tecnologia assistencial de modo a processar documentos adequadamente. As implementações do OpenXML podem ser muito pequenas e fornecem funcionalidade centralizada ou podem abranger todo o conjunto de recursos. Os mecanismos de capacidade de extensão incorporados ao formato garante espaço para inovação.

A padronização da especificação dos formatos e sua manutenção ao longo do tempo garantem que várias partes podem confiar seguramente nele, confiantes de que desenvolvimentos futuros vão se beneficiar das verificações e dos equilíbrios permitidos por um processo de padrões abertos. Existe a urgente necessidade de um padrão para formatos de documentos abertos que seja capaz de manter os bilhões de documentos que foram criados nos formatos pré-existentes e os bilhões que continuam a ser criados a cada ano. Os avanços tecnológicos em hardware, redes e uma infra-estrutura de software baseada em padrões tornam isso possível. A diversificação explosiva na demanda do mercado — incluindo significativos investimentos já feitos em sistemas comerciais de missão crítica — torna isso essencial.

7 CITAÇÕES

1. **Ecma International.** TC45 – Formatos do Office Open XML. *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 (Fourth Edition). *World Wide Web Consortium*. [Online] 2006.
<http://www.w3.org/TR/2006/REC-xml-20060816/>.
6. **W3C**. Namespaces in XML 1.0 (Second Edition). *World Wide Web Consortium*. [Online] 2006.
<http://www.w3.org/TR/2006/REC-xml-names-20060816/>.
7. **XBRL International**. XBRL Specifications. *Extensible Business Reporting Language*. [Online]
<http://www.xbrl.org/Specifications/>.
8. **Health Level Seven**. HL7 ANSI-Approved Standards. *Health Level Seven*. [Online]
http://www.hl7.org/about/directories.cfm?framepage=/documentcenter/public/faq/ansi_approved.htm.
9. **W3C**. W3C Web Services Architecture. *World Wide Web Consortium*. [Online] 2002.
<http://www.w3.org/2002/ws/>.
10. **W3C**. W3C XML Signature. *World Wide Web Consortium*. [Online]
<http://www.w3.org/Signature/>.
11. **W3C**. W3C XML Encryption. *World Wide Web Consortium*. [Online] 2001.
<http://www.w3.org/Encryption/2001/>.
12. **W3C**. XSL and 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>.