

連載

知って得する

土木技術者のための

Information Technology
IT

講座



土木学会 土木情報システム委員会 情報共有技術小委員会編

第2回 構造化に基づくドキュメント交換・共有技術

昨今、インターネットの爆発的な流行、電子メールの普及など、情報の流通経路や扱われ方が大きく変化しています。産業界でも、“e-ビジネス”などと称し、ビジネススタイルの改革に迫られている企業も少なくないと思われます。

その中で、彗星の如く現れ、これからの情報処理の仕組みをますます大きく変えようとしている“XML (エックスエムエル: eXtensible Markup Language)”。今回は、このXMLに代表される構造化したドキュメントについて、なるべく簡単に説明していきたいと思います。

ドキュメントって何?

普段、何気なく使っている“ドキュメント”という言葉ですが、あらたまって“ドキュメントとは何か”と問われたとき、さまざまなイメージが思い浮かび、説明に困ってしまうのではないのでしょうか。

建設産業においては、工事申請書や報告書、設計図や参考書籍などの設計図書、場合によっては写真や映像などが、さまざまな形態で、いろいろな立場の人を、いささか氾濫気味に、とどまることなくやり取りされています。

これらを“ドキュメント”とすれば、

「ある事柄についての内容や情報を記録、または、他の人に伝達するために用いられるもの」とでも言い表せるのでしょうか。語源をたどると、ラ

テン語の documentum (docIre [教える] + mentum [手段])、“教える手段”という意味の言葉だそうです。

ここでは、企業内や企業間、行政機関など取り交わす、いろいろな媒体に表現された情報全般を、広く“ドキュメント”にとらえ、その中で、電子的な媒体で扱われているものを、便宜上勝手ながら、“ドキュメント情報”と言うこととします。

ドキュメントの3成分

現在のドキュメントの多くは、様式文書とされています。文書を構成する成分を抽出すると、図-1のように、内容、構造、体裁の3つに分類できます。

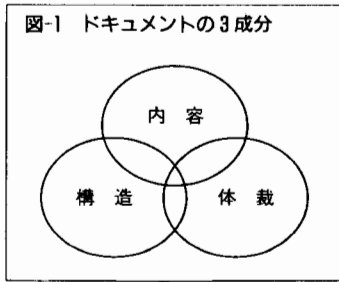


図-2 簡単な文書例

2000年4月1日

委員各位
 「第1回 小委員会開催案内」
 発行：小委員会事務局
 日時：2000年4月20日 14時～
 場所：土木学会 A会議室
 議題：1. 設立主旨説明
 2. 活動計画
 備考：終了後、懇親会を予定。

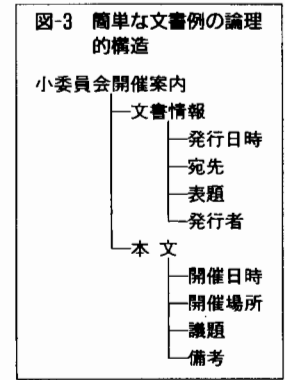


図-2 に、ごく簡単な文書の一例を示しました。一見して、おおむね内容は理解できると思いますが、会議の開催案内です。事務局から各委員へ、開催日時や場所、議題などを伝達するための文書です。この文書で3つの成分を確認していきましょう。

まず、第1の成分の“内容 (コンテンツ)”は、記録の内容や伝達したい情報を綴った、文書や図・表そのものです。図-2のすべてが“内容”といえます。

第2の成分の“構造 (ストラクチャ)”は、文書の構成上の位置づけや、一つひとつの要素の関係を表現したものです。日ごろ、無意識に扱われることが多く、ちょっとわかりづらいかもかもしれませんが、図-2の文書の“日時：”の欄に注目してください。文書全体を見れば、その欄は、“第1回小委員会の開催日時”を表現していることがわかります。しかし、“日時：”の欄だけしか読まなければ、その日時が、この文書の作成日時なのか、または発行日時なのか、何の日時を指しているのか理解できないでしょう。つまりこの日時は、文書の記載順序、前後関係などから無意識に“開催日時”と判読されています。

あらかじめ“この部分は<小委員会開催案内>の<開催日時>です”とわかるように書いておけば、部分的に読んだとしても、意図された内容が的確に伝わるでしょう。このように文書の構成要素

と各要素の関係を表現しているのが“構造”です。

図-3 は、図-2の文書例の構造を表現したものです。この文書は、表題などの前提条件を表現している文書情報と、議題など伝達事項である本文で構成されていることがわかります。

最後の成分である“体裁 (スタイル)”は、読み手によりわかりやすく、的確に意思伝達できるよう、修飾方法などを指定する成分です。図-2の文書例では、この文書が開催案内であることを強調するために、表題をセンタリングし、字体や文字の大きさを変えたり、発行日時や発行者欄を右寄せにして、文書の構造を判読しやすく工夫しています。

これから説明するXMLは、これら3つの成分を分離して扱い、第1の成分である“内容”に、第2、第3の成分である“構造”や“体裁”に関する情報を、“タグ”という形式で与えることにより、文書を記述します。

タグ付け (マークアップ) 文書の歴史

ここで、“タグ”という言葉が出てきました。

タグ (tag) とは、英語で“荷札”の意味ですが、文書の“タグ”について、少し詳しく見ましょう。

文書のタグ付け (マークアップ) の起源は、コンピュータで文書が扱われるずっと昔、印刷技術

図-4 マークアップ文書の例

<p>■タイプセット用命令コードを含む文書例</p> <p><右揃え><明朝><10 pt>平成〇年〇月〇日 <左揃え><ゴシック><10 pt>委員各位 <センタリング><太ゴシック><12 pt>「第1回 小委員会開催案内」 <右揃え><明朝><10 pt>発行：小委員会事務局</p> <p>■論理構造のマークアップを施した文書例</p> <p><作成日時>平成〇年〇月〇日 <宛先>委員各位 <表題>「第1回 小委員会開催案内」 <発行者>発行：小委員会事務局</p>
--

が発達する中、印刷原稿の作成にタイプセット(植字機)と呼ばれる機械が使われるようになったところにさかのぼります。印刷の主流が、金属の活字を版に並べ印刷される活版印刷から、写真製版に変化していった時代です。文字の大きさや書体の変更といった操作を記録した機械への命令文書を、印刷する文書の先頭に付けることで、印刷の自動化が実現できるようになりました。その後も開発が続き、単なる命令コードから、複数の種類の機械で共通に扱え、また、可読性を高めるために、文書の論理構造を記述するように進展していきました(図-4)。

さらに、コンピュータにより文章の処理が行われるようになり、印刷技術と離れてより汎用化され、ドキュメント記述言語として用いられるようになります。やがて、1986年、ISO(国際標準化機構)により、国際規格として“SGML(エスジーエムエル：Standard Generalized Markup Language)”が制定されるまでに発展しました。

SGMLの制定=マークアップ言語の標準化

当初、SGMLが制定された目的は、さまざまな様式で作られているドキュメント情報を、利用者のハードウェアやソフトウェア環境の相違にかかわらず、相互に利用可能な記述方法として確立することにありました。そこで、コンピュータの

種類やソフトに依存しない、文書の構造そのものを記述する言語標準として制定されました。

SGMLは、DTD(デーティーデー：Document Type Definition：文書型定義)と呼ばれる文法で、文書の論理的構造をタグ("<"と">"で囲まれた文字)として定義し、次に、内容の記述の前後に、その定義されたタグを付与する方法で記述されます。それから、必要に応じて体裁の情報を表すタグが定義され、付与されます。

SGMLでは、多種多様なドキュメントの記述が可能であった反面、ネットワークでの利用が考慮されていませんでした。また、厳格性を重視したばかりに規格が複雑で利用者も限られ、「静かなる革命」と皮肉った言われ方もされたようです。

HTMLの登場=情報のネットワーク化

SGMLに変わり、登場後、インターネットの普及を加速させながら、急速に広まっていったのが、HTML(エイチティーエムエル：Hyper Text Markup Language)です。皆さんの中にも、企業のホームページを閲覧したり、航空機の時刻を調べて予約をしたりと、すでに日常的にインターネットを活用されている方も多いと思います。

実は、ホームページはこのHTMLにより記述された文書でできています。この文書がインターネットに接続されたいろいろなコンピュータの中

表-1 SGMLとHTMLの比較

	SGML	HTML
設計 コンセプト	<ul style="list-style-type: none"> ・文書の構造表現が主題 ・異機種間の情報交換が目的 	<ul style="list-style-type: none"> ・オンライン型の情報共有が目的 ・操作性を重視
利用形態	<ul style="list-style-type: none"> ・利用方法は定義されていない ・人とシステムとの対話形式 	<ul style="list-style-type: none"> ・インターネット(Webサーバとブラウザ)での利用が前提 ・人とシステムとの対話形式
特性	<ul style="list-style-type: none"> ・国際標準 (ISO) 規格である ・タグの追加や構造定義が可能 (あらゆる用途に向く) ・HTMLやPDFなどへ変換が可能 	<ul style="list-style-type: none"> ・全世界で1つのタグセット ・操作が容易 ・安価に利用できる
抱える課題	<ul style="list-style-type: none"> ・インターネット登場前の規格 ・ネットワークでの活用が考慮されていない ・複雑で修得が困難 →利用アプリケーションも複雑になってしまう ・利用コストが高い 	<ul style="list-style-type: none"> ・文書構造を持っていない ・タグの追加が困難 ・大量で複雑な情報集積や高度な情報処理に不向き ・長期にわたる情報の保存(保全)が困難

におかれていて、インターネットに接続されたパソコンで閲覧しているだけです。ウェブブラウザと呼ばれるアプリケーションを使って、マウスのクリックだけで次から次へと、簡単に膨大な情報を閲覧できるハイパーテキストというしくみが作られ爆発的に普及したのです。次々と機能も強化され、検索機能やアプリケーション構築技術も導入され、ますます便利になりました。

HTMLはSGMLを母体として作られていますが、インターネットで情報を流通させることが目的で、機能面の豊富さよりも、操作の容易性が追求されました。“タグ”に関しては、以下の点でSGMLと大きく異なっています。

- ▷ HTMLのタグは、ブラウザで情報が閲覧されるときに体裁情報に限定される(文書の構造的要素は持てない)
- ▷ タグの機能はあらかじめ決められていて、世界共通である
- ▷ ユーザーによる自由なタグの追加や拡張は許されない

このような機能によって、爆発的に普及した反面、複雑なデータ処理には不向きで、情報の維持・管理が大変であるなど、課題も生じてきました。引き続きバージョンアップも行われていますが、全面的な解決は難しいと言われ、登場したのがXMLです。

XMLによる改革の始まり

XMLの説明に入る前に、もう一度、SGMLとHTMLの特性を表-1に整理しておきました。

どちらの言語も、当初の目的は達成できたものの、さらなる拡張が困難なところまでたどりついてしまったようです。

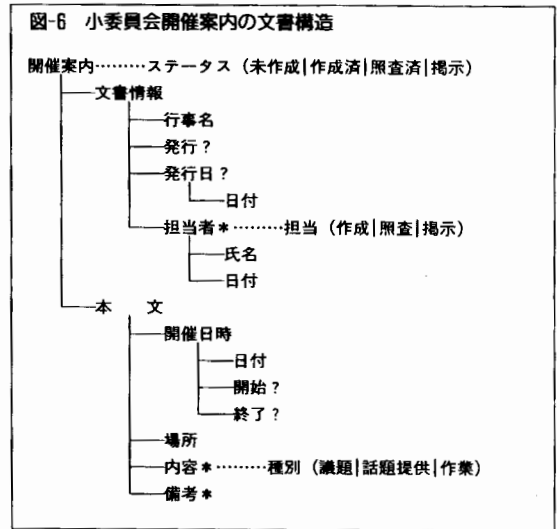
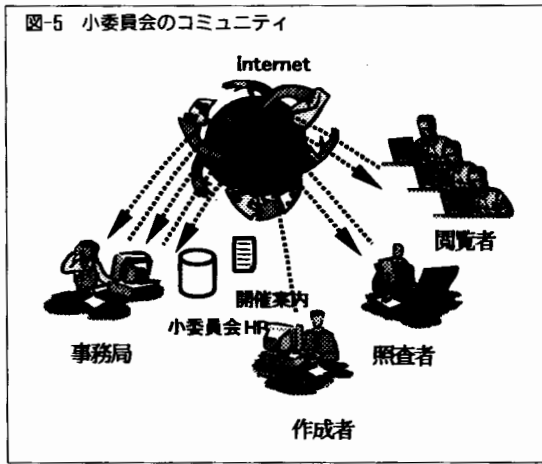
そこで、1996年、インターネットの規則を定めているW3C(ダブリュースリーシー: World Wide Web Consortium)において、XMLのワーキンググループが発足し、1998年、わずか2年という速さで、XMLのバージョン1.0が制定され公表されました。

XMLは、以下のコンセプトで設計されました。

- SGMLとの互換性を維持し、HTMLのように扱いやすい
- HTMLによる文書の構造表現の限界を打破し、より高度な交換・共有環境を構築
- システム同士の情報交換を実現(人とのインターフェイスは、利用者のソフトウェアに委ねる)
- 文書の内容と構造、体裁を完全に分離

この結果、XMLは以下のような優れた特性を備えています。

- ① 国際標準に準拠
- ② ネットワークとの親和性が高い
- ③ 拡張性が高い(タグに関する制約がない)
- ④ 情報の再利用が容易



⑤ 大規模な情報蓄積と管理が可能

このような背景から、XML は非常に高い評価を得て、凄まじい勢いで広まりつつあります。次々に新しい関連規格も追加され、XML を専門に扱う企業が設立されたり、さまざまなアプリケーションの開発が着手されるなど、話題が絶えないといった状況で、今後ますます発展し続けることでしょう。

XML 記述説明と適用事例紹介

XML について、登場の背景と特性を説明してきましたが、だいたい理解していただけたと思います。次に、XML を使った、文書の具体的な記述方法について説明しましょう。

XML 文書は、複数の人々での利用が前提になります。文書の構造を設計するためには、その文書に対し、どんな人がどのように関わるのかというコミュニティ（共同体）を把握することが重要になります。

現在、私たちの小委員会での情報交換には、ほぼ 100% インターネットが利用されています。図-5 のように、事務局にコンピュータを設置し、ホームページを維持・管理しながら、情報の発信・共有・保管を行っています。必要に応じ役割分担を行い、協力し合って委員会活動が運営されています。

会議の開催案内を例にとると、小委員会の開催が決まったら、まず、作成担当が開催案内を作成し、ホームページに登録します。同時に、担当者に内容が正しいか照査依頼が通知されます。承認後公開ページとして掲示され、参加者に通知されます。必要に応じ、この手順の進行状態を確認できる機能も必要でしょう。最後に、各参加者が出欠の返事を行い、会議開催予定の伝達という一連の作業が完了します。このようなコミュニティで利用される、文書の構造は 図-6 のようになります。

図-1 の文書例に、多少必要な項目を肉付けしました。発行日や担当者といった要素の後に“?”や“*”を付けてあります。これらは、要素の登場回数を表現しています。図中、点線で要素とつながれている情報は、要素の持つ詳しい情報（属性）です。この文書の状態をあらわす“ステータス”や、作成担当や照査担当などの“担当者”は複数登場するので、それらの違いを属性という形で表現しています。

文書の構造が設計できたら、いよいよ XML の作成です。XML 文書は、内容、構造、体裁の 3 つの成分に対応する、XML、DTD、XSL の 3 つの異なるファイルからなり、XML ファイルは、“XML 宣言”“文書型宣言”“XML インスタンス”の 3 つの部分で構成されます。“XML 宣言”では、XML のバージョンや、使用する文字コー

ドなど日本語で書かれたXMLの文書であることを宣言します(省略も可能です)。

“文書型宣言”では、文書中の要素、属性、実体(文書中に埋め込む外部ファイルの指定など)を定義します。この中の構造成分として定義される文書型定義がDTDです。

“XMLインスタンス”は残りの部分で、文書の内容をタグ付き文書として記述される部分です。

リスト-1は、図-1の文書をXMLで記述した文書です。

```

リスト-1 XML記述事例(開催案内.xml)

<?xml version="1.0" encoding="Shift_JIS"?> ..... ①
<?xml-stylesheet type="text/xsl" href="開催案内.xsl"?> ..... ②
<!-- 文書型宣言 --> ..... ③
<!DOCTYPE info SYSTEM "開催案内.dtd"> ..... ④

<!-- XMLインスタンス -->
<info status="publish"> ..... ⑤
  <header> ..... ⑥
    <title>第1回 小委員会</title>
    <publisher>小委員会事務局</publisher>
    <publication>
      <date>
        <year>2000</year>
        <month>4</month>
        <day>1</day>
      </date>
    </publication>
    <staff role="drafting">
      <name>
        <name1>山田</name1>
        <name2>太郎</name2>
      </name>

```

```

</date>
  <year>2000</year>
  <month>3</month>
  <day>25</day>
</date>
</staff>
<staff role="approved"> ..... ⑦
  ***** 中 略 *****
</staff>
</header>
<body> ..... ⑧
  <open>
    <date>
      <year>2000</year>
      <month>4</month>
      <day>20</day>
    </date>
    <start>
      <time>
        <hour>14</hour>
      </time>
    </start>
    <open>
      <place>土木学会 A会議室</place>
      <content>設立趣旨説明</content>
      <content>活動計画</content>
      <note>終了後、懇親会を予定</note>
    </body> ..... ⑧'
  </info> ..... ⑤'

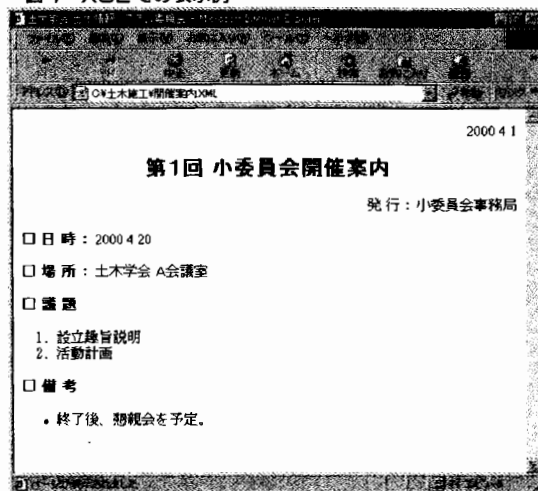
```

最初の2行、①と②がXML宣言です。①は、XMLのバージョンが1.0、Shift JISコードで記述されていることを、②は、体裁情報に“開催案内.xsl”というファイルを用いることをそれぞれ宣言しています。③の“<!--”と“-->”で挟まれた部分は、コメントとして無視されます。④は、文書型宣言で、最初の要素名がinfoで、文

表-2 要素の出現回数

要素名のみ	必ず1回出現
要素名?	0 もしくは1回出現
要素名+	1回以上出現
要素名*	0回以上出現

図-7 XSL での表示例



書型定義として、“開催案内.dtd”というファイルを取り込むよう宣言しています。

⑤から下が XML インスタンスです。開始タグ (<...>)、内容、終了タグ (</...>) の順に記述しますが、階層構造をとる場合は、親の要素の中に記述され、最終行⑤’に info の終了タグがあります。タグの中の status="Publish"では、status 属性に値 Publish を代入しています。⑥からが header (文書情報) 要素です。⑦で2つ目の staff 要素を記述しています。⑧から⑧’が、body (本文) 要素です。

次に、ここで使われているタグを定義している、DTD の記述を見てみましょう (リスト-2)。

```

リスト-2 XML DTD 記述事例(開催案内.dtd)

<ELEMENT info (header, body)> ..... ①
<ATTLIST info status (non | drafting | approved |
publish) "non">
..... ②
<ELEMENT header (title, publisher?, publication?, staff*)>
<ELEMENT title (#PCDATA)> ..... ③
<ELEMENT publisher (#PCDATA)>
<ELEMENT publication (date)>
<ELEMENT staff (name, date)>
<ATTLIST staff role (drafting | approved |
publish) "drafting">

<ELEMENT body (open, place, content*, note*)>
<ELEMENT open (date, start?, end?)>
<ELEMENT start (time)>
<ELEMENT end (time)>
<ELEMENT place (#PCDATA)>
<ELEMENT content (#PCDATA)>
<ATTLIST content type (subject | topics | work) "subject">
<ELEMENT note (#PCDATA)>
    
```

```

<ELEMENT date (year, month, day, oftheweek?)>
<ELEMENT year (#PCDATA)>
<ELEMENT month (#PCDATA)>
<ELEMENT day (#PCDATA)>
<ELEMENT oftheweek (#PCDATA)>
<ELEMENT time (hour, minute?)>
<ELEMENT hour (#PCDATA)>
<ELEMENT minute (#PCDATA)>
<ELEMENT name (name1, name2)>
<ELEMENT name1 (#PCDATA)>
<ELEMENT name2 (#PCDATA)>
    
```

DTD では、要素型宣言、属性リスト宣言、実体宣言、記法宣言が行われます。リスト-2の①のように“<!ELEMENT 要素名 内容モデル>”と記述されるのが要素型宣言です。階層化された構造や、文字列の内容を持つ要素は、③の title 要素のように内容モデルに“(#PCDATA)”と記述します。XML 文書では、要素の出現順序と出現回数が制御されます (表-2)。

②は、属性リスト宣言です。“<!ATTLIST 要素名 属性名 属性値の候補 規定値 “>”と記述します。

リスト-3 は、リスト-2の DTD とリスト-1の XML 文書を 図-7 のように表示させるための、XSL (eXtensible Stylesheet Language) の記述例です。

```

リスト-3 XSL 記述事例(開催案内.xsl)

<?xml version="1.0" encoding="Shift_JIS"?>
    
```

```

<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<HTML>
<HEAD>
<TITLE>土木学会 土木情報システム委員会</TITLE>
</HEAD>
<BODY>
<P ALIGN="RIGHT">
<xsl:value-of select="info/header/publication/date"/></P>
<H2 ALIGN="CENTER">
<xsl:value-of select="info/header/title"/></H2>
<P ALIGN="RIGHT">発行 :
<xsl:value-of select="info/header/publisher"/></P>
<B>□ 日 時 : </B>
<xsl:value-of select="info/body/open/date"/><BR/><BR/>
<B>□ 場 所 : </B>
<xsl:value-of select="info/body/place"/>
<P><B>□ 議 題</B></P>
<OL>
<xsl:for-each select="info/body/content">
<LI STYLE="font-family:MS 明朝;font-size:12pt">
<xsl:value-of select="."/></LI>
</xsl:for-each>
<OL>
<P><B>□ 備 考</B></P>
<UL type="disc">
<xsl:for-each select="info/body/note">
<LI STYLE="font-family:MS 明朝;font-size:12pt">
<xsl:value-of select="."/></LI>
</xsl:for-each>
</UL>
</BODY>
</HTML>
</xsl:template>
</xsl:stylesheet>

```

XSL は、HTML や CSS (Cascading Style Sheets) といったホームページの技術と混在して利用することもできます。

XML の功績と展望

XML に関して、基本的な部分をかいつまんで、駆け足で説明してきました。内容・構造・体裁の3つの成分が、それぞれ独立して扱われているのが理解していただけたでしょうか。

構造化され、整然と記述された情報というのは、その情報の細部まで理解でき、情報の交換・共有を行う場合にも、非常に有益です。

ドキュメント記述言語として XML を説明しましたが、XML の用途は、ドキュメント記述にとどまりません。今後、XML による情報の記述が進むにつれ、データベース連携や、エージェント(自律した制御ができるプログラム)による自動的な情報交換も行われ、新しい企業連携や、商構造の変化が起こると言われています。まだまだ課題も多くありますが、建設産業の文書情報やインターネット上の膨大な情報が、効率よく再利用できる日を目指して、技術革新に取り組んでいます。

(執筆：中村真一、皆川勝、宮本勝則、馬場一秋)

《参考文献》

- 1) 菊田昌弘：XML がもたらすもの、bit、Jan 1999、p.52-62
- 2) つくりながら学ぶ XML、OpenDesign、1999年12月号