

# XML 入門

XML部会(A)

## メンバー紹介

部会長	神矢 正一	太平工業（株）大分支店 電気計装部ITソリューショングループ
副部会長	岩切 宏幸	三井造船システム技研（株） カスタマアプリケーションサービス事業部 第3開発部
	荒木 靖博	新日鉄情報通信システム（株） システム第1部
	角 輝久	（株）シーエイシー システム部
	吉岡 秀樹	三菱商事太陽（株） 統括室
	遠藤 豊紀	（株）オーイーシー ビジネスシステム営業課
	松崎 ゆかり	（株）オーイーシー システムサービス第1課
	戸高 志郎	（株）オーイーシー 東京支社 システム営業課
	荒金 博	大分交通（株） 情報通信営業部
	肌野 文昭	（株）アトムス事務器 ソフト開発課

## はじめに

最近、XML (Extensible Markup Language) という言葉がさまざまなメディアで取り上げられ、注目されている。XMLは開発されて間もない新しい言語だが、eコマース(電子商取引)への利用など、さまざまな場所での応用が期待されている。

そこで我々は、この次世代標準マークアップ言語に注目し、XMLの特徴及び現在の利用状況を全員で確認し、さらに理解を深めるために、XMLを使用したシステムを製作した。また、今後、我々の業務にXMLがどのように活かせるのかを調査・研究した。

本書は、我々がXMLの研究活動を通して理解して来たことを「XML入門」書としてまとめたものである。

### - 活動の経緯 -

#### 平成 12 年 7 月 12 日

- ・ 部長・副部会長の選出
- ・ 活動テーマについての話し合い

#### 平成 12 年 7 月 19 日

- ・ 活動テーマ「XML 入門」に決定
- ・ HTML の学習(XML と HTML の違いを知るため)

#### 平成 12 年 8 月 8 日

- ・ XML について各自調査した内容を報告・質疑応答

#### 平成 12 年 8 月 23 日

- ・ 共通教材による学習(参考文献：日経ソフトウェア 9 月号)

#### 平成 12 年 9 月 13 日

- ・ 学習後の不明点の討議・検討

#### 平成 12 年 9 月 27 日

- ・ 論文の項目・テーマの選定

#### 平成 12 年 10 月 11 日、25 日、11 月 14 日、29 日、12 月 13 日、26 日

- ・ XML 調査グループとシステム製作グループに分かれ、目的を絞って活動

#### 平成 13 年 1 月 17 日、26 日、2 月 2 日

- ・ 論文作成、校正

## 目次

1 . X M L の生い立ちと背景 .....	5
2 . X M L の特徴 .....	6
3 . X M L を取り巻く技術 .....	1 0
4 . X M L の有効活用 .....	1 3
4 - 1 E D I と X M L .....	1 3
4 - 2 行政機関の X M L 対応 .....	1 5
4 - 3 マスメディアの X M L 対応 .....	1 6
4 - 4 携帯端末と X M L .....	1 7
5 . X M L の製作例 (顧客管理システム) .....	1 8
5 - 1 システム構築の目的 .....	1 8
5 - 2 実行環境 .....	1 8
5 - 3 システム構成 .....	1 9
5 - 4 アプリケーションの役割 .....	2 0
5 - 5 顧客データのツリー構造 .....	2 0
5 - 6 ソースデータ内容説明 .....	2 1
5 - 6 - 1 Customer . X M L 構成 .....	2 1
5 - 6 - 2 ソースプログラム説明 .....	2 2
6 . X M L の今後の課題 .....	2 9
7 . まとめ .....	3 0

## 1. XMLの生い立ちと背景

なにかと話題になっているXML。まず、XML登場の背景をみてみよう。

インターネットでの情報検索・サービスシステムのひとつであるWWW（World Wide Web）は数人の核物理学者の実験プロジェクトとして始まった。Webによって世界中の読者に向けて、お金をかけずに簡単に電子文書を配布できるというメリットと、HTMLというWebブラウザ上でテキストを表示するためのタグ付け言語が開発されたことにより爆発的な成長をとげた。（HTML以外にも言語は存在するが、最も普及したのがHTMLであり、HTMLの存在によりWebが成長したといっても過言ではない）

Webの普及にともない、様々なWebアプリケーションへの要求もまた増加していった。そして、それらの要求を満たすにはHTMLでは限界があることが明らかになった。

シンプルなHTMLとは対照に汎用的な言語としてSGMLがある。SGMLは言語を記述するための言語「メタ言語」でありHTMLもその応用例である。SGMLは非常に強力で基幹システムを支えられる技術であるが、大規模で複雑そして高価である。習得が困難で高コストであるため、SGMLは費用対効果の面で魅力がないことがわかっている。

このような背景を元に登場したのがXML (Extensible Markup Language)である。XMLは、Webでの利用を目的にSGMLを縮小して生まれた規格であり、Webへの対応に必要な機能が新たに追加されている。

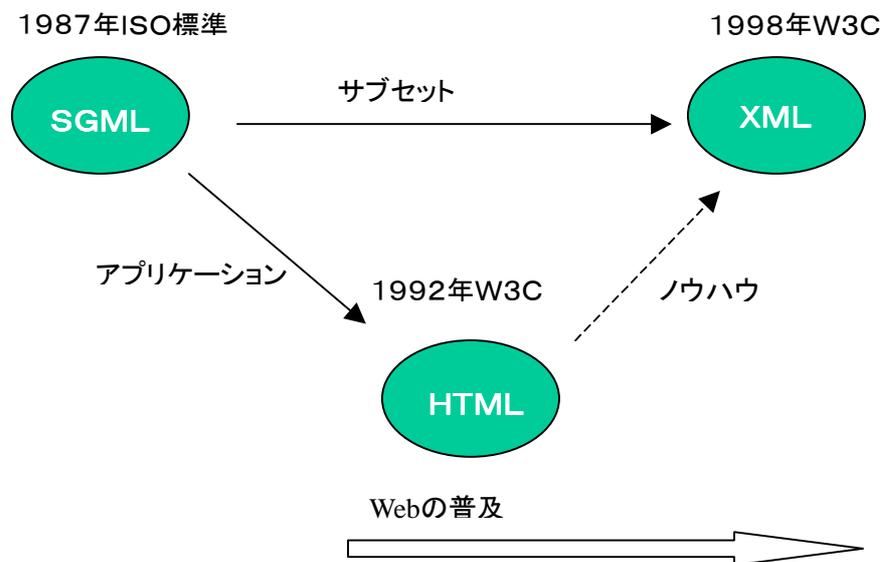


図1 XMLの生い立ち

XMLは、W3C（the World Wide Web Consortium）の賛助の下で1996年に構成されたXMLワーキンググループによって開発され、1998年2月10日にXML 1.0がW3C勧告となった。

注) W3Cとは、WWWで用いられる技術の標準化と推進を目的とする国際学術研究機関。

## 2 . X M L の特徴

X M L は S G M L をベースに、H T M L で得られたノウハウを取り入れて開発されている。そのために両者の長所を合わせ持っている。また、新たな機能も追加されている。では、その特徴をみていこう。

### タグ名を自由に設定できる

おそらく、X M L の最大の特徴は S G M L から受け継いだ「タグ名を自由に設定できる」ということであろう。これは「情報に意味をもたせることができる」ということである。そのデータが何のデータであるのか、人間だけでなくコンピュータにも判断できるようになった。

その効果を H T M L と比較して情報検索の点から例をあげてみよう。カタログのページに重量や金額といった数値が載っていたとしよう。

ここで金額が 1000 未満のものを検索することを考える。H T M L では数値の意味をコンピュータでは判断できない。人間が目で見えて判断するしかないであろう。では、X M L の場合はどうだろう。重量は<重量>、金額は<金額>というタグを作って内容に意味を持たせてやればよい。コンピュータは数値の入っているタグ名が何であるかによってそれが金額であることを判断できるわけである。

注) H T M L と比較して特徴を述べているが、これは、X M L と H T M L そのものの優劣を言っている訳ではない。例えて言うならば H T M L は文書を記述する Word のようなワープロソフト、X M L はデータを管理する Access のようなデータベースソフトである。Access は Word と比べるとデータを管理できるという特徴をもっていると言っているようなものである。

### データとスタイルの明確な分離

情報としての X M L ドキュメントと表示形式としての X S L ドキュメント(スタイルシート)を明確に分離することで、情報の再利用性を高めている。1つのデータを複数の画面で使う、ワンソースマルチユースが容易となる。

H T M L では情報と表示形式が一緒になっているので、同じ情報を相手に応じてレイアウトを変えたい場合、相手の数だけ H T M L が必要になる。X M L では相手に応じた X S L を用意してやればよい。X M L を更新すれば全てに反映されるのでデータの更新が

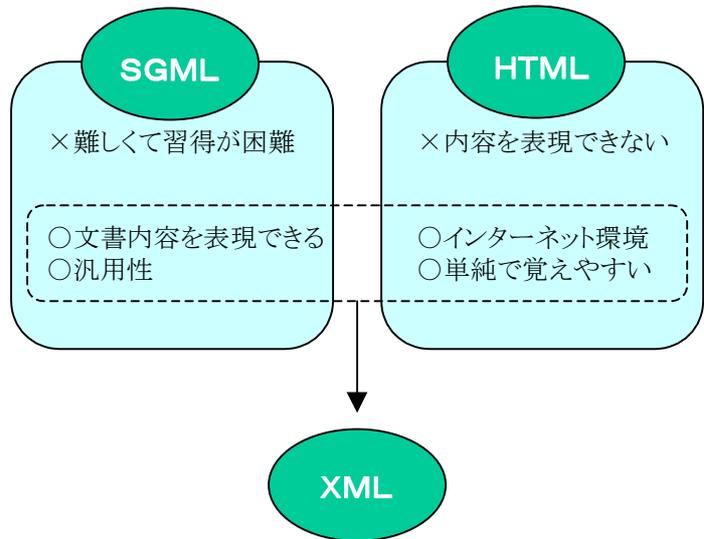


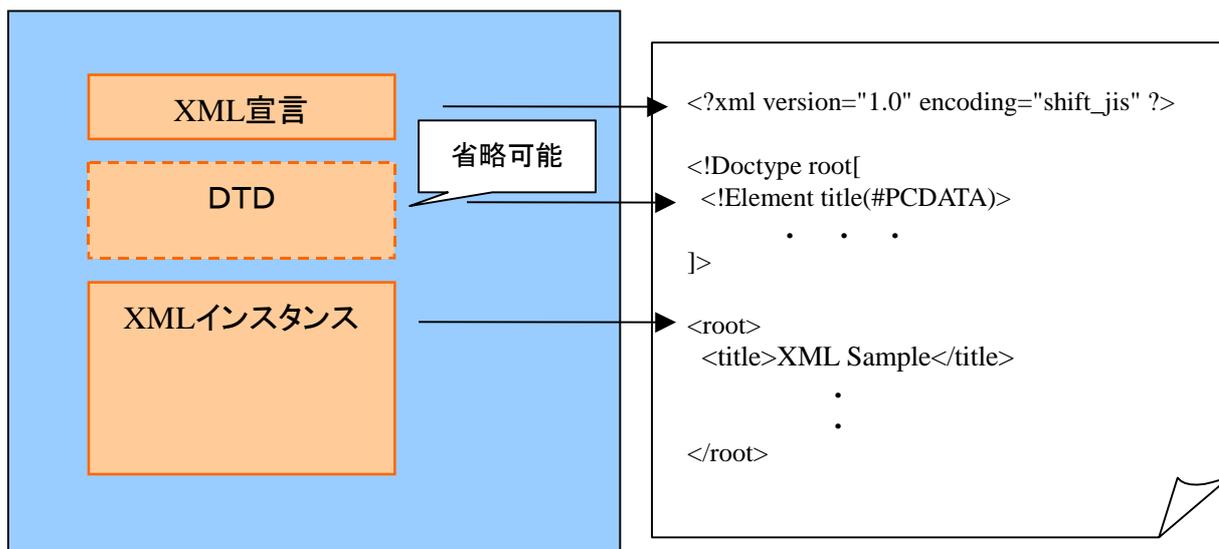
図2 1 X M L の特徴

容易となる。

また、Web 以外のシステムにデータを流用するのも容易になり、データの再利用が徹底できる。

テキストデータである

XMLはテキストデータで、プラットフォームやアプリケーションに依存しないデータ交換が可能。OSを気にすることもなくデータの扱いが容易である。



UNICODE による国際化

XMLはUNICODEを採用している。また、XML対応ソフトウェアにもUNICODEのサポートを義務付けられている。そのため、UNICODEを使用すれば、文字コード系の食い違いによる文字化けを起こさずに各国固有の文字が取り扱える。

プログラムから利用できる

DOMやSAXといったXMLにアクセスするためのインターフェースがある。これらのインターフェースは言語に依存しないのでさまざまなアプリケーションからXML文書の内容を操作できる。

データ交換に利用できる

XMLはデータ交換に利用するにも適している。一般的に普及しているCSVと比べると柔軟性が高い。タグによって項目名が付いているのでデータの順番を変えてもプログラムに影響しない。ただし、タグのデータ量が多くなるケースが発生する。

同名のタグを区別する仕組みを持つ（名前空間）

異なる文書を一緒に使うというようなことがXMLでは可能である。ここで困るのが同じ名前のタグであるのに異なる意味で使用している場合である。<名前>というタグで、

あるXMLでは人名。また、あるXMLでは地名という使い方をされていると混乱が生じてしまう。そこで考えられたのが名前空間(Namespace)である。名前空間をタグ名の前に付ける事でそのタグ名が誰の作ったどういう構造のものが区別できるようになる。名前空間にはURIを使用しているため、名前空間が重なることはない。

次にXMLの構造を見てみよう。

ツリー構造(階層構造)である  
 ツリー構造は高度な表現が可能であり、人間に理解しやすい形式である。  
 マニュアルで言えば、章があり節があり本文がある。また、本文の中に箇条書きがあるといった形である。

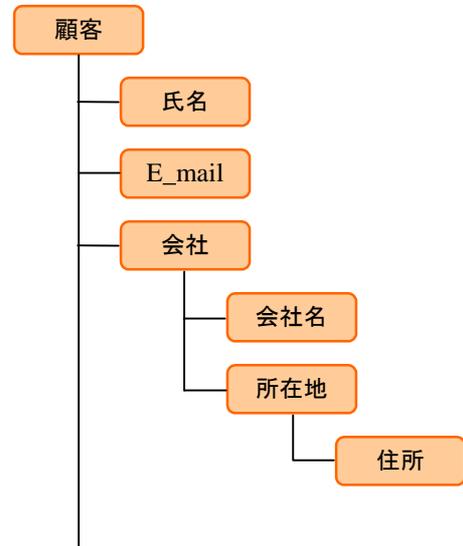


図 2 2 ツリー構造の例

XMLはHTMLに比べて記述が厳格である

表 2 1 XMLとHTMLの記述の比較

項目	HTML	XML
入れ子構造	タグの順番がおかしくてもブラウザ側で対応可能	入れ子の構造(タグの順番)に厳格
タグ名の大文字小文字	区別しない	区別する
閉じタグ	必ずしも必要ないタグがある	必ず記述する。ただし、<タグ名

今まで述べた特徴を踏まえ、XML、SGML、HTMLの比較を以下にまとめる。

表2 2 XML、SGML、HTMLの比較

比較項目	XML	SGML	HTML	備考
電子閲覧	○	△	○	SGML仕様には文字の大きさや配置等WEB上での見えかたを指定するスタイル機能がない
印刷	○	△	×	HTMLには印刷ページ概念がない
Web適合度	◎	×	◎	SGMLはURLリンクを仕様としてもたない
文書交換	◎	◎	×	HTMLは情報を明確に構造化表現できない
再利用性	◎	◎	×	情報部品としての有効性、情報属性の明確性
データベース	◎	◎	×	データベースの親和性
リンク対応	◎	×	○	ハイパー・リンク仕様の保持
管理の容易性	○	○	×	コンテンツ管理
作成の容易性	○	△	○	作成上の操作性
オブジェクト	◎	○	△	データ・オブジェクト、オブジェクト・プログラミング
プログラミング	◎	○	△	データ処理の自動化、アプリケーション・プログラム開発の容易性

◎:最適 ○:適用可 △:適用可、機能不足 ×:不向きまたは無関係

### 3 . X M L を取り巻く技術

X M L 文書は、バージョン、文字エンコーディングなどを宣言する X M L 宣言、文書の定義を行う D T D、タグのついたデータの集まりである X M L データとで成り立っている。

<b>X M L 宣言</b>	<code>&lt;xml version="1.0" encoding='shift-jis' ? &gt;</code>
<b>D T D</b>	<code>&lt;!DOCTYPE 情報&gt;</code> <code>&lt;!ELEMENT 名前(#PCDATA)&gt;</code> <code>&lt;!ELEMENT 住所(#PCDATA)&gt;</code>
<b>X M L データ</b>	<code>&lt;情報&gt;</code> <code>&lt;名前&gt;名前花子&lt;/名前&gt;</code> <code>&lt;住所&gt;大分県大分市 X X X 町&lt;/住所&gt;</code>

図 3 - 1 は、パーサーが X M L ファイルと D T D ファイルを読み込んで解析し、W e b ブラウザや D B に書き込むという処理の仕組みを示したものである。

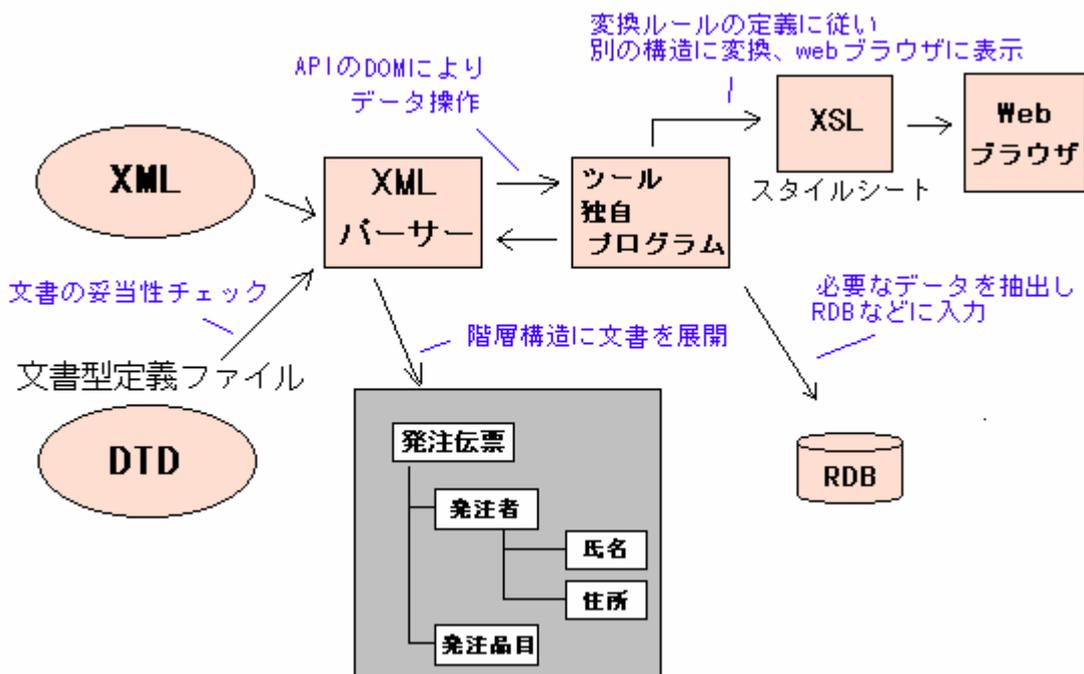


図 3 - 1 X M L を処理する仕組み

では、XML文書进行处理する仕組みを述べる。

XML文書は、通常テキストファイルとして存在している。これを、XMLパーサーが階層構造に解釈し、他のアプリケーションで利用可能な形にする。つまり、パーサーを介すことにより、データを操作することができる。

DTDファイルとは、XML文書の取り得る構造を記述したXMLのスキーマ言語の一種であり、XML文書の正しさがある程度まで自動的にチェックする。

DTDは、スキーマ言語としての機能のほかに実体参照などの定義も含んでおり、スキーマ言語以外の機能を利用される場合もある。DTDは、XMLの前身であるSGMLより基本的な構文を受け継いでいるため、他のXML関連技術と比較して、長い利用実績があり、長所も短所も明確になっている。

一方で、DTDで満たせないニーズのために、他のスキーマ言語の開発も盛んである。しかし、実体参照などはDTDと関連するため、スキーマ言語としてDTD以外を使用する場合でも、DTDの知識は必要とされる。

他のアプリケーションで利用可能な形になったオブジェクトは、パーサーの標準化されたAPI (Application Programming Interface : コンピュータとプログラムの橋渡しの規約) のDOMとSAXを用いて目的の要素を取り出す。

ここで、上記のAPIについて述べる。

#### DOM (Document Object Model)

DOMとは、XMLパーサーにアクセスするためのAPIの一つであり、W3Cが公式に公開した唯一のAPIである。

DOMは、XML文書を「DOMツリー」と呼ばれるツリー構造として扱う為、XMLパーサーがXML文書全体を読み込んだ後でなければ、文書内のデータにアクセスすることができない。

また、DOMツリーは通常メモリ上に要素オブジェクトのツリーを作成する為、大規模なXML文書进行处理する場合には、その分メモリ容量も要求される。

その代わりに、DOMを利用すれば、XML文書内のデータの順番に関係なくアクセスできる。

#### SAX (The Simple API for XML)

SAXは、DOMと異なり、文書を先頭から順に読み込んでいき、そこで発生したイベントを経由して情報をアプリケーションプログラムに伝達する。

DOMの場合はXML文書全体の構文解析が終了し、DOMツリーができるまではアプリケーションプログラムは何の情報も得ることはできないが、SAXの場合は解析中に分かったことから順次伝達する。

プログラムやドキュメント構造に制約が伴う場合もあるが、構文解析の終了まで待たず

に処理を開始できるというドキュメントを読み込みながらの処理なので、反応も早くメモリ消費の少ないプログラムを作成することもできる。

Web で表示するためのスタイルシートについて述べる。

X S L （ Extensible Stylesheet Language ）

X M L 専用の、 X M L ドキュメント内のどの部分のデータをどのように表示するかといった表示形態を記述したスタイルシートを記述するマークアップ言語に、 X S L がある。

X M L に従い、取り出したドキュメントを W e b ブラウザに表示する。

また、 X S L は、 X M L を H T M L に変換したり、別の D T D の X M L に変換することもできる。

X S L は、 2 0 0 0 年 1 1 月時点で、勧告候補となっている。

## 4 . X M L の有効利用

X M L を採用すると、” データ交換のインフラにインターネットが使える ” 既存の業務システムとの連携が容易 ” といった利点がある。

### X M L の位置づけ

X M L 有効利用方法として、企業間はもちろん、企業内 LAN や WAN で接続したシステム間で送受信するデータの標準形式としても注目されている。X M L により、複数のシステム間で EDI ( 電子データ交換 ) を実現するための開発コストを抑えたり、やり取りするデータの項目などの変更にかかる負荷を軽減できる。

### 4 - 1 E D I と X M L

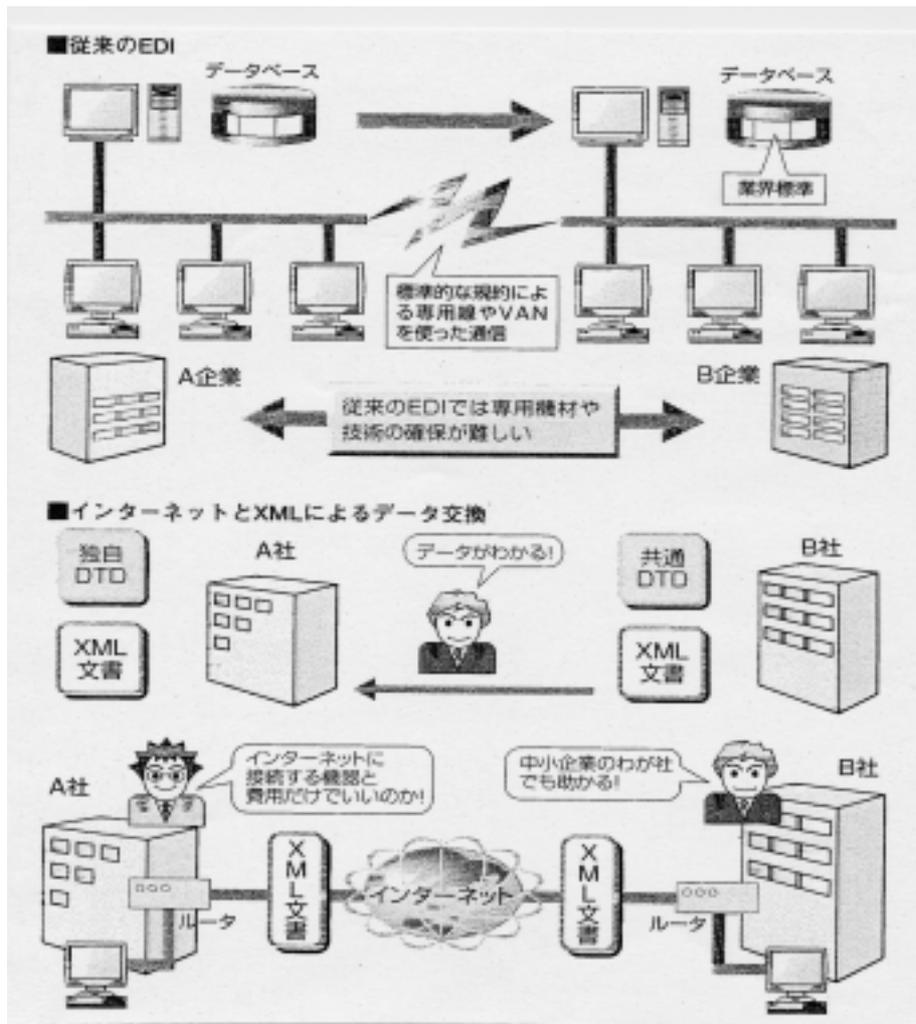
特にインフラとして安価なインターネットを利用できるので、企業間でやり取りするデータの形式をにするとコスト削減の効果が大きい。それは従来型の EDI ( 電子データ交換 ) では、一般にインフラとして高価な専用線や VAN ( 付加価値通信網 ) を利用し、データの形式は業界ごとに固有のものを使うことが多い。それに対し XML のデータは、Web と同じ HTTP ( ハイパーテキスト・トランスファ・プロトコル ) で交換できるからである。

また、最近インターネット上で HTML 形式のデータ交換を行う「Web-EDI」も普及し始めている。安価なインターネットを使える点では、これも有効なデータ交換方法である。しかし、もともと HTML は Web ブラウザへの画面表示用の言語なので、データ交換に最適な言語とは言えない。このため、HTML データの送受信と基幹系システムの入出力との間に、人手による処理が必要な場合が多い。XML を使った EDI では、基幹系システムと連携してデータを自動処理できる。これは、XML がデータの構造や属性を自由に定義できる言語だからである。

XML はデータ自体とは別に構造や意味を定義できるので、データ項目の変更も容易である。例えば、EC ( 電子商取引 ) サイトで扱う商品の追加・削除に伴ってデータ項目を変更する場合、作業の負荷を軽減できる。これは、XML を社内システムのデータ形式として採用した場合にも有効だと思われる。

例えば、これまで EDI を導入することが難しかった中小企業などでもインターネットと XML 文書を利用し、電子データの交換が考えられる事になる。

また、EDI で交換されるデータがほかの目的に再利用しにくいことに比べ、XML 文書によるデータはさまざまなアプリケーションによって別の目的に利用することが簡単になっている。ほかの企業から受け取ったデータを自社内のデータに再利用することができる。



以上を要約すると、現状良く使われている個々独自のデータ・フォーマットではなくXMLを使うことで、以下の利点が得られると考えられる。

- (1) 各システム間の連携が容易になる。
- (2) 将来どちらかのツールを別製品に入れ替えても他方に影響しない(柔軟性が高い)。
- (3) 他のシステムやツールと連携させるニーズが出たとき、それらがXMLにさえ対応すれば連携が可能(拡張性が高い)。

## 4 - 2 . 行政機関のXML対応

行政・法律・会計などの分野では多くの紙文書を抱え、文書管理が欠かせないものになっている。特に行政機関などでは、従来からSGMLが採用されており、表4-1のようなSGMLにしたがった各種DTDなどが制定されている。

表4-1 行政機関によるSGMLのDTD

省庁	DTD
総務庁	電子公文書DTD
総務庁	白書DTD
特許庁	特許HTML
厚生省	医療用医薬品の添付文書DTD

(省庁名はH12.12現在の名称です)

このような紙文書を多く抱える行政機関では、XML文書による文書管理が注目されている。表4-1のSGMLのDTDや、各種申請書などのファイルをXML文書で統一する試みが考えられている。

また、現在建設省による工事用のデジタル写真を管理するためのタグなどを始め、XMLにしたがったDTD案が検討されており、今後はその有効利用と展開が期待されている。

以下は、実際の使用例で、XMLの利点を生かす為に、行政の電子化に向けて各種の申請書をXML化してインターネット環境で運用するための実験システムの事例である。平成11年の4月から平成12年の3月までの期間、ニューメディア協会のもと各ベンダーが参加し、XML電子申請システムの開発及び実証実験が行われた。

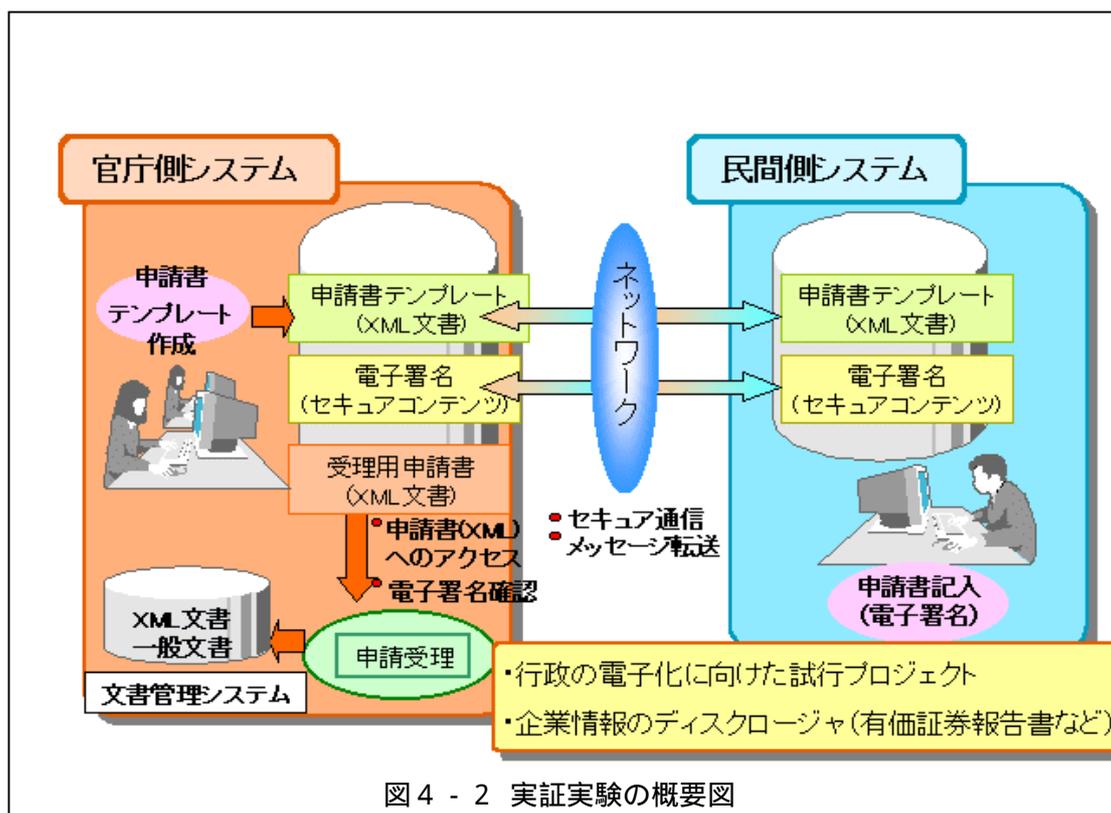


図4-2 実証実験の概要図

### 4 - 3 . マスメディアのXML対応

S G M L が従来から利用されている分野に出版がある。事典・辞書など長く改訂を重ねる出版の分野では、特にXMLによる文書管理が必要とされている。

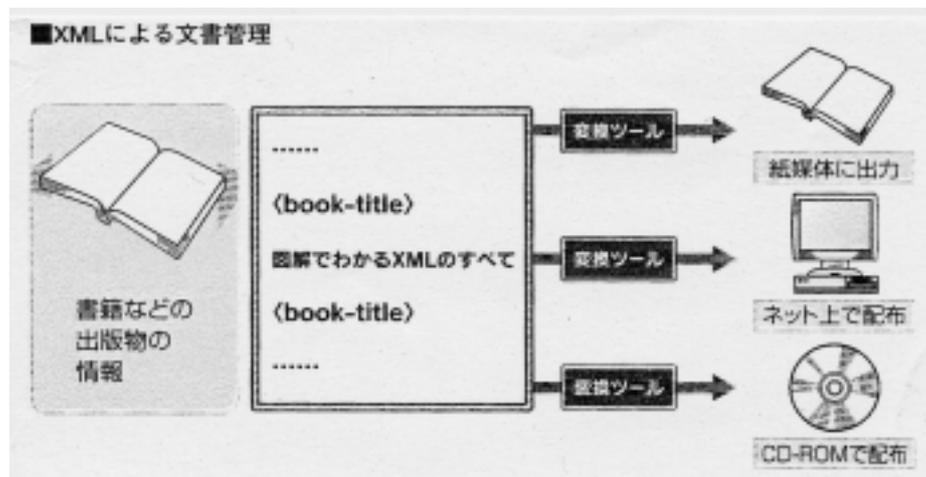
出版物に関する情報をXML化することによって、紙媒体ばかりでなくさまざまなメディアに対応することができるようになる。

たとえば、アドビ・システムズ社によって開発されたP D F (Portable Document Format)として出版物をインターネット上で配布することが可能になる。

また、CD-ROMなどの媒体に収録して出版物を配布することも可能となる。

XML文書として管理されていれば、どのようなデータ形式にも変換して利用していくことが容易になる。

このような分野では、書籍などの出版物の構造を表すためのタグを決めて、XML文書化することが考えられている。



#### 4 - 4 . 携帯端末とXML

現在では、携帯電話やカーナビ、ゲーム機などさまざまな携帯端末からインターネット上の文書や画像にアクセスする機会が増えている。

携帯端末は、以下の問題があるため、XML文書を使い、小型の端末に適した書式情報を与えて表示する方法が有用となる。

ハードウェアの性能がデスクトップパソコンに比べ低い。

(CPU処理能力が劣る、画面サイズが小さい)

ネットワークへの接続速度が高速とはいえないため、ダウンロードに時間がかかったり、小さな画面では全体を表示出来ない。

さらに携帯端末の特徴を生かしたXML文書の利用も考えられている。

たとえば、現在地理上の位置情報や道路情報・道案内情報などに関する標準的なDTDを作成する試みがある。携帯端末にとって有用な情報をXML文書として携帯端末で入手することで、便利なシステムを利用することができるようになりそうである。官庁や民間団体では、このようなさまざまな情報をXMLで標準化することが進められている。

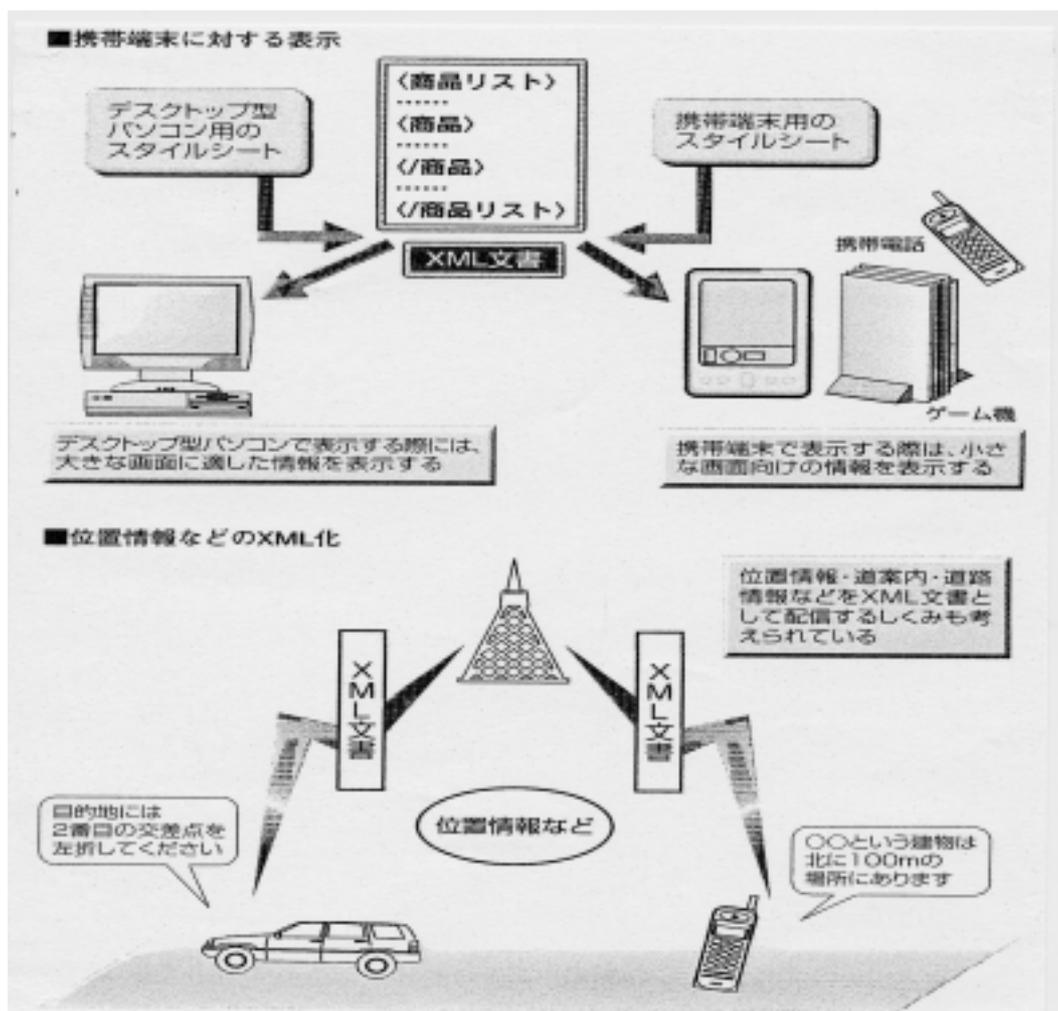


図4 - 4 携帯端末とXML

## 5 . X M L の製作例 ( 顧客管理システム )

### 5 - 1 . システム構築の目的

RDB の代わりに XML を使用する。

XML、XSL の記述方法の習得。

X M L パーサー (Microsoft XML パーサー "msxml.dll") の使用方法の検証。

ワンソース・マルチユース の検証。

### 5 - 2 . 実行環境

#### (1) OS

Windows95、Windows98、WindowsNT4.0

#### (2) ブラウザ -

Microsoft Internet Explorer (IE) 5.0

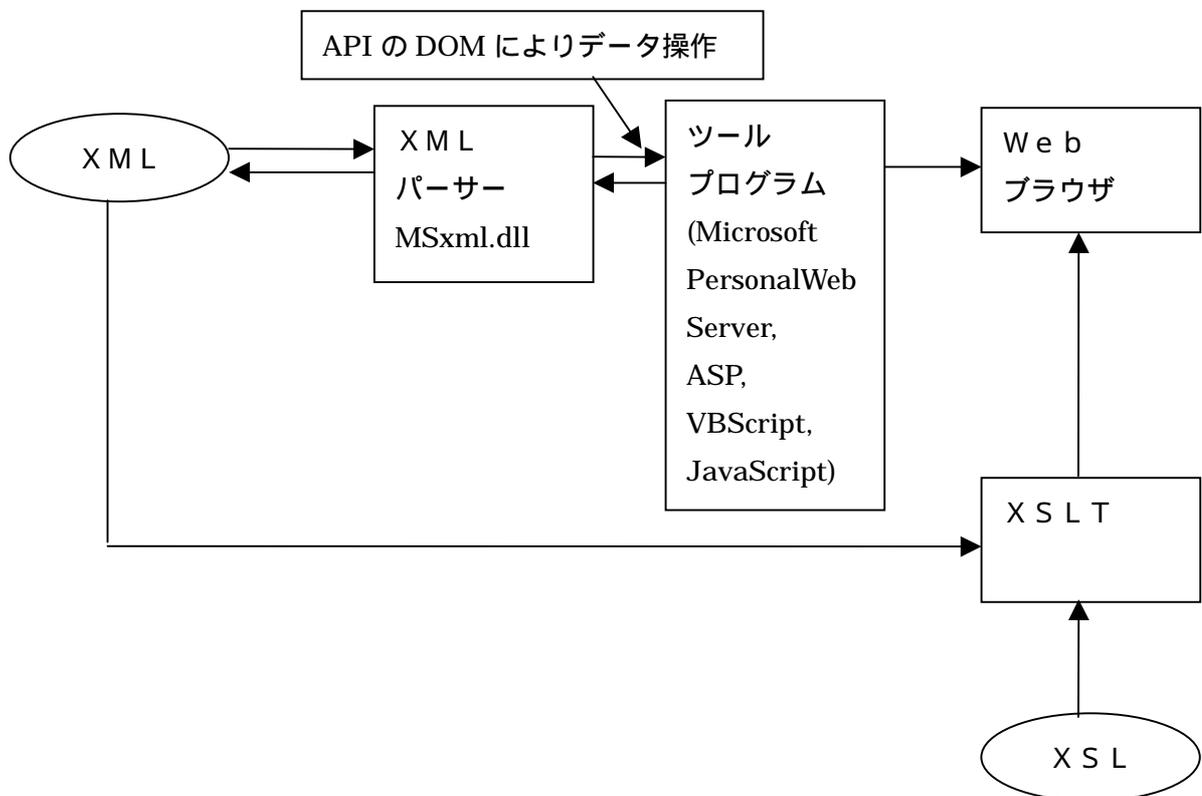
Microsoft XML パーサーを標準搭載

#### (3) Web サーバー

Microsoft PersonalWebServer (NT の場合は、IIS)

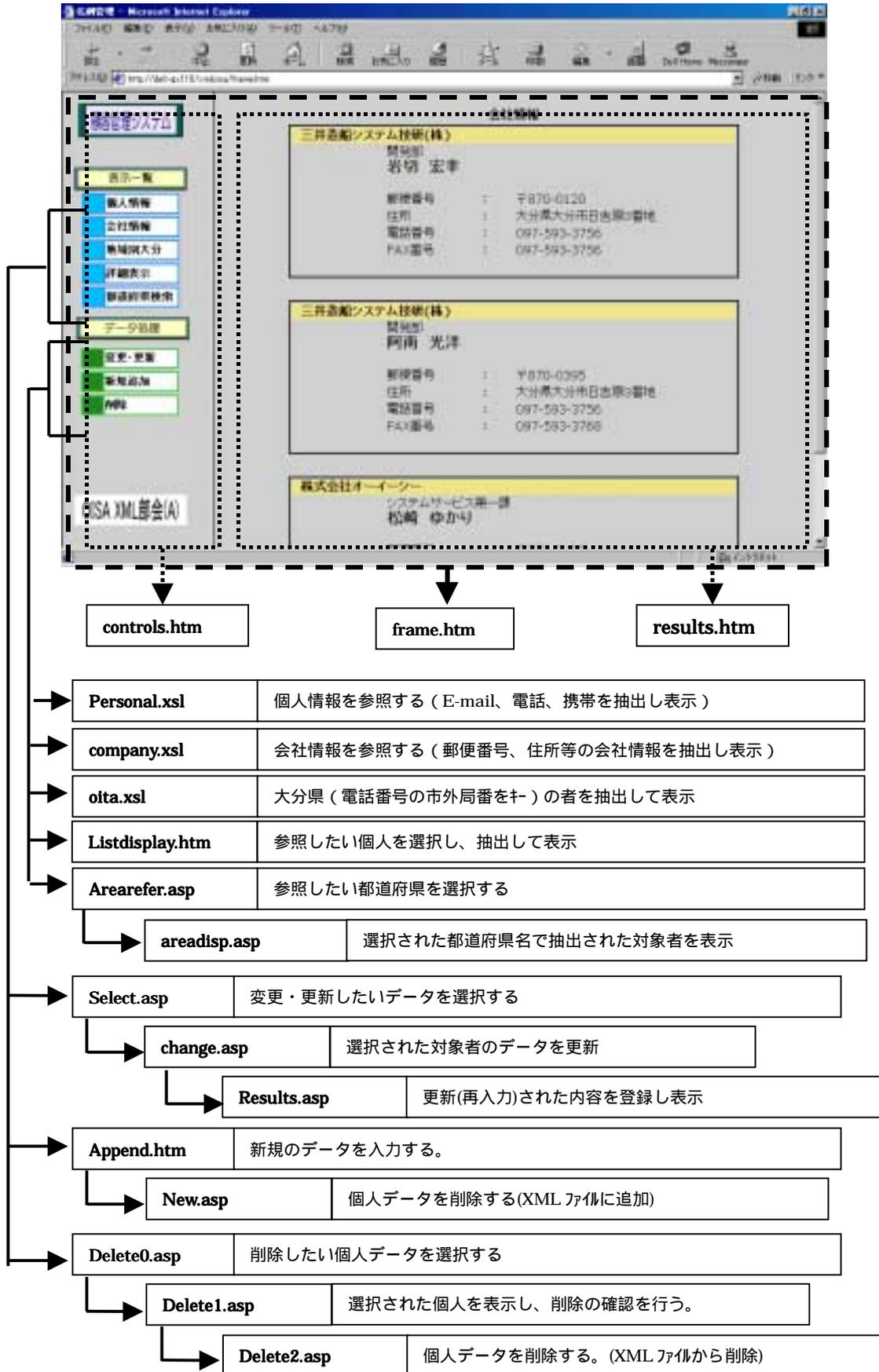
#### (4) プログラム作成言語 (スクリプト言語)

- ASP ( Active Server Page )
- VB Script
- Java Script



### 5 - 3 . システム構成

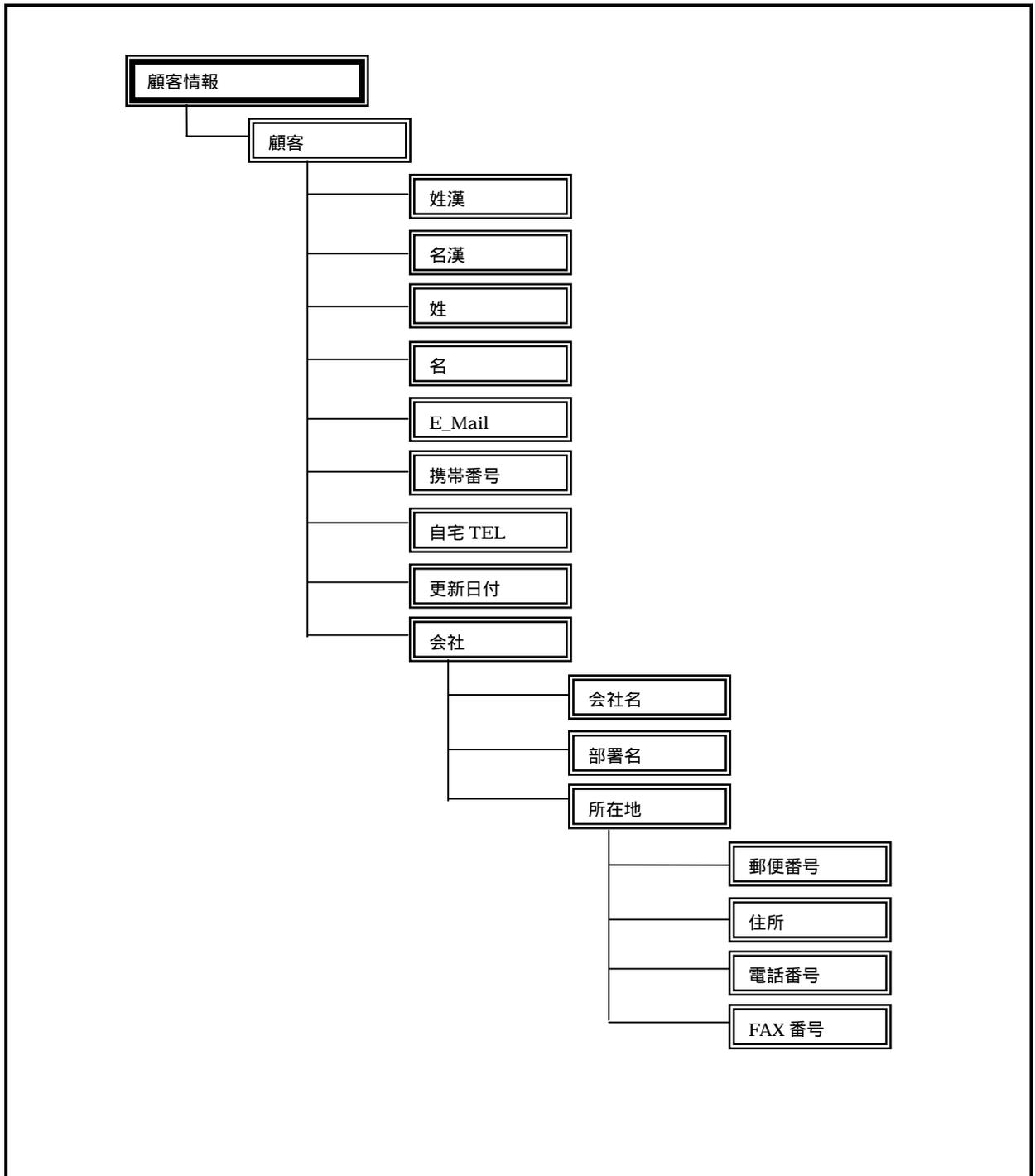
[ 画面・処理構成 ]



#### 5 - 4 . アプリケーションの役割

- XML : 業務処理データを格納
- HTML : 静的な画面表示に使用
- XSL : 変更が頻繁に起こる画面表示に適用
- CSS : 背景のデザインを格納
- ASP : データ処理のロジックを記述

#### 5 - 5 . 顧客データのツリー構造(Customer.XML)



## 5 - 6 . ソース・データ内容説明

### 5 - 6 - 1 . Customer.XML 構成 ( データ部分 )

```
<顧客>
  <姓漢>阿南</姓漢>
  <名漢>光洋</名漢>
  <姓>アナン</姓>
  <名>コウヨウ</名>
  <E_mail>kanan@msr.mes.co.jp</E_mail>
  <携帯番号>090-0001-0001</携帯番号>
  <自宅TEL>097-531-0002</自宅TEL>
  <更新日付>01/01/22</更新日付>
  <会社>
    <会社名>三井造船システム技研(株)</会社名>
    <部署名>開発部</部署名>
    <所在地>
      <郵便番号>870-0395</郵便番号>
      <住所>大分県大分市日吉原 3 番地</住所>
      <電話番号>097-593-3756</電話番号>
      < F A X 番号>097-593-3768</ F A X 番号>
    </所在地>
  </会社>
</顧客>
```

## 5 - 6 - 2 . ソースプログラム説明

### personal.xsl (個人情報参照)

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

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
```

```
<xsl:template match="/">
```

XSL テンプレートを適用するノードを定義

```
<table width="75%" align="center">
```

```
<tr>
```

```
<th width="100%">個人情報</th>
```

```
</tr>
```

```
</table>
```

制御式により繰返出力を行う

```
<xsl:for-each select="顧客情報/顧客">
```

```
<TABLE STYLE="border:2px solid black" width="80%" align="center"
```

```
border="3">
```

```
<TR STYLE="font-family:Verdana; font-size:12pt; padding:0px 6px">
```

```
<td width="30%">
```

```
<table width="100%">
```

```
<tr>
```

```
<TD align="center">
```

```
<font size="2">
```

```
<xsl:value-of select="./姓" />
```

```
<xsl:value-of select="./名" />
```

```
</font>
```

```
</TD>
```

```
</tr>
```

```
<tr>
```

```
<TD align="center">
```

```
<B>
```

```
<font size="4">
```

```
<xsl:value-of select="./姓漢" />
```

```
<xsl:value-of select="./名漢" />
```

```
</font>
```

```
</B>
```

```
</TD>
```

```
</tr>
```

```
</table>
```

```
</td>
```

```
<td>
```

```
<table width="100%">
```

マッチングした結果を生成

```

<TR STYLE="font-family:Verdana; font-size:12pt; padding:0px
    6px">
  <TD width="30%">E-mail</TD>
  <TD width="5%"> : </TD>
  <TD width="65%">
    <xsl:value-of select="./E_mail" />
  </TD>
</TR>
<TR STYLE="font-family:Verdana; font-size:12pt;
    padding:0px 6px">
  <TD width="30%">携帯</TD>
  <TD width="5%"> : </TD>
  <TD width="65%">
    <xsl:value-of select="./携帯番号" />
  </TD>
</TR>
<TR STYLE="font-family:Verdana; font-size:12pt;
    padding:0px 6px">
  <TD width="30%">自宅</TD>
  <TD width="5%"> : </TD>
  <TD width="65%">
    <xsl:value-of select="./自宅TEL" />
  </TD>
</TR>
</table>
</td>
</TR>
</TABLE>
<br />
<br />
</xsl:for-each>
</xsl:template>
</xsl:stylesheet>

```

## Company.xsl (会社情報参照)

```

<?xml version="1.0" encoding="shift_jis" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
  <xsl:template match="/">

```

```

<table width="75%" align="center">
  <tr>
    <th width="100%">会社情報</th>
  </tr>
</table>
<xsl:for-each select="顧客情報/顧客/会社">
  <TABLE STYLE="border:2px solid black" width="75%" align="center">
    <TR STYLE="font-family:Verdana; font-size:12pt; padding:0px 6px">
      <TD colspan="4" STYLE="background-color:Khaki"
        width="70%">
        <b>
</TD><xsl:value-of select="./会社名" />
</TR>
<TR STYLE="font-family:Verdana; font-size:12pt; padding:0px 6px">
  <td width="15%"></td>
  <TD colspan="3">
    <xsl:value-of select="./部署名" />
    <br />
    <b>
      <font size="4">
        <xsl:value-of select="./姓漢" />
        <xsl:value-of select="./名漢" />
      </font>
    </b>
    <br />
  </TD>
</TR>
<TR STYLE="font-family:Verdana; font-size:12pt; padding:0px 6px">
  <td width="15%"></td>
  <TD width="15%">郵便番号</TD>
  <TD width="5%"> : </TD>
  <TD>
    〒
    <xsl:value-of select="./所在地/郵便番号" />
  </TD>
</TR>
<TR STYLE="font-family:Verdana; font-size:12pt; padding:0px 6px">
  <td width="15%"></td>
  <TD width="15%">住所</TD>
  <TD width="5%"> : </TD>

```

```

<TD>
  <xsl:value-of select="./所在地/住所" />
</TD>
</TR>
<TR STYLE="font-family:Verdana; font-size:12pt; padding:0px 6px">
  <td width="15%"></td>
  <TD width="15%">電話番号</TD>
  <TD width="5%"> : </TD>
  <TD>
    <xsl:value-of select="./所在地/電話番号" />
  </TD>
</TR>
<TR STYLE="font-family:Verdana; font-size:12pt; padding:0px 6px">
  <td width="15%"></td>
  <TD width="15%">FAX 番号</TD>
  <TD width="5%"> : </TD>
  <TD>
    <xsl:value-of select="./所在地/F A X 番号" />
  </TD>
</TR>
<TR>
  <TD colspan="4"></TD>
</TR>
</TABLE>
<br />
<br />
</xsl:for-each>
</xsl:template>
</xsl:stylesheet>

```

## listdisplay.htm ( 詳細データ表示 )

```

<html>
<head>
<script language="javascript">

var objxml;
var objxsl;
var card = new Array();

```

```
var num;
```

```
/* listdisplay.htm が初めに読み込まれたときにロードする関数 */
```

```
function init()
```

```
{
```

```
    dispList(); //個人名リストの表示
```

```
    table.style.display = "none"; //初めは詳細表示のテーブルを非表示にする
```

```
}
```

```
/* 個人名リストの作成と表示 list.xml を使用 */
```

```
function dispList()
```

```
{
```

```
    objxml = new ActiveXObject("Microsoft.XMLDOM");
```

```
    objxsl = new ActiveXObject("Microsoft.XMLDOM");
```

```
    objxml.async = false;
```

```
    objxsl.async = false;
```

```
    objxml.load("CUSTOMER.XML");
```

```
    objxsl.load("list.xml");
```

```
    //id="ListDisplay"に XSL を使って表示
```

```
    ListDisplay.innerHTML = objxml.transformNode(objxsl);
```

```
}
```

```
/* 個人名リストから一人選択した後、詳細情報を表示する関数 */
```

```
function getSelect()
```

```
{
```

```
    num = document.listForm.select.selectedIndex; //個人名リストの上から何番目が選択されたか
```

```
objxml = new ActiveXObject("Microsoft.XMLDOM");  
objxsl = new ActiveXObject("Microsoft.XMLDOM");  
objxml.async = false;  
objxsl.async = false;  
  
objxml.load("CUSTOMER.XML");  
objxsl.load("list.xml");
```

XML コメント、XSL ファイルを  
"DOM" のオブジェクトとして読み  
込む

```

//配列 card にタグごとの詳細情報を入れる
card[0] = objxml.getElementsByTagName("姓漢").item(num).text + " " +
        objxml.getElementsByTagName("名漢").item(num).text;
card[1] = objxml.getElementsByTagName("姓").item(num).text + " " +
        objxml.getElementsByTagName("名").item(num).text;
card[2] = objxml.getElementsByTagName("更新日付").item(num).text;
card[3] = objxml.getElementsByTagName("E_mail").item(num).text;
card[4] = objxml.getElementsByTagName("携帯番号").item(num).text;
card[5] = objxml.getElementsByTagName("自宅 T E L").item(num).text;
card[6] = objxml.getElementsByTagName("会社名").item(num).text;
card[7] = objxml.getElementsByTagName("部署名").item(num).text;
card[8] = objxml.getElementsByTagName("郵便番号").item(num).text;
card[9] = objxml.getElementsByTagName("住所").item(num).text;
card[10] = objxml.getElementsByTagName("電話番号").item(num).text;
card[11] = objxml.getElementsByTagName(" F A X 番号").item(num).text;

```

```

for(i=0;i<12;i++)
{
    document.all("顧客").item(i).innerHTML = card[i];
}
table.style.display = "block"; //詳細表示を ID=table の下に展開
dispList(); //個人名リストの再表示
}

```

選択された個人の情報を XML  
ドキュメントから DOM のノードを  
使用して取得する

```
</script>
```

```
</head>
```

```
<body onload="init();" bgcolor="#FFF5EE">
```

```
<center>
```

```
<h2>詳細表示</h2>
```

詳細表示する個人名を選んで「詳細表示」ボタンを押してください。<br><br>

```
<FORM name="listForm">
```

```
    <DIV ID="ListDisplay"></DIV><br>
```

```
    <INPUT TYPE="button" VALUE="詳細表示" onclick="getSelect()">
```

```
</FORM>
```

```
<TABLE ID="table" BORDER="1" CELSPACING="0" CELLPADDING="4" WIDTH="350"
FRAME="box" RULES="none" BGCOLOR="#FFFFFF">
```

```
    <TR BGCOLOR="#FFE4B5"><TD width="20"></TD>
```

```

        <TD><font size="+1"><B><SPAN ID=" 顧客 "></SPAN>      <SPAN ID=" 顧客
"></SPAN></B></font></TD></TR>
        <TR><TD width="20"></TD>
        <TD align="right">LAST UPDATE : <SPAN ID="顧客"></SPAN></TD></TR>
        <TR><TD width="20"></TD><TD>
        <TABLE BORDER="1" CELLSPACING="0" CELLPADDING="4" WIDTH="280" FRAME="void"
RULES="rows">
        <TR><TD WIDTH="60">E-mail</TD><TD><SPAN ID="顧客"></SPAN></TD></TR>
        <TR><TD>携帯 TEL</TD><TD><SPAN ID="顧客"></SPAN></TD></TR>
        <TR><TD>自宅 TEL</TD><TD><SPAN ID="顧客"></SPAN></TD></TR>
        <TR><TD>勤務先</TD><TD><SPAN ID="顧客"></SPAN></TD></TR>
        <TR><TD>所属部署</TD><TD><SPAN ID="顧客"></SPAN></TD></TR>
        <TR><TD>所在地</TD>
        <TD>〒<SPAN ID="顧客"></SPAN><br><SPAN ID="顧客"></SPAN></TD>
        </TR>
        <TR><TD>TEL</TD><TD><SPAN ID="顧客"></SPAN></TD></TR>
        <TR><TD>FAX</TD><TD><SPAN ID="顧客"></SPAN></TD></TR>
        </TABLE>
        </TD></TR>
</TABLE>
</center>
</body>
</html>

```

## list.xml ( 選択された個人を抽出 )

```

<?xml version="1.0" encoding="shift_jis"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">

<xsl:template match="/">
<select name="select" size="10">
<xsl:for-each select="//顧客">
    <xsl:element name="option">
        <xsl:value-of select = "./姓漢"/>
        <xsl:value-of select = "./名漢"/>
    </xsl:element>
</xsl:for-each>
</select>
</xsl:template>
</xsl:stylesheet>

```

XMLドキュメントから1レコードずつ  
抽出し、氏名を表示する。

## 6 . X M L の今後の課題

X M L 文書は、データベースWWWを連携したシステムを構築・運用する場合など、さまざまなパターンで使われることが予想される。X M L 文書が実際にこれらの業務システムに取り入れられ、普及していくために欠かせないのが共通のタグの取り決めである。

X M L では、タグを独自に取り決めることができる。共通のタグを企業間で取り決めておくと、自社内や企業間でX M L 文書を取り入れたシステムを構築することができ、業務の効率化につながる。

共通のタグを企業間で取り決めることにより、企業間取引やデータの効率的な利用が期待できる。このため、各業界毎に目的に応じた独自のタグを取り決め、D T D を作成する試みが進んでいる。また、その部分でコンセンサスをとっておかないと、誰も使わない仕様だけが残ることになる。

主なものとして、表6 - 1のような仕様が各業界で検討されている。

表6-1 XMLの仕様例

DTD	名称	仕様
XHTML	eXtensible Hyper Markup Language	HTMLをXMLで定義した規格
MathML	Mathematical Markup Language	数学関連の式を記述するための規格
VoiceXML	Voice Extensible Markup	音声関連の規格
CDF	Channel Data Format	マイクロソフト社のチャンネル用規格
RDF	Resource Description Framework	メタデータを記述するための規格
CML	Chemical Markup Language	化学関連の式を記述するための規格
CBL	Common Business Library	コマース・ワン社の電子商取引に関する規格
cXML	Commerce XML	アリバ社の電子商取引に関する規格
SVG	Scalable Vector Graphics	ベクタ図形記述用の規格
MML	Medical Markup Language	電子カルテに関する標準規格
SMIL	Synchronized Multimedia Integration Language	音楽や画像
WML	Wireless Markup Language	PDA、携帯端末用の規格
HDML	Handheld Device Markup Language	PDA、携帯端末用の規格
JepaX	Japan Electronic Publishing Association	日本電子出版協会電子出版物刊行用の規格

## 7. まとめ

我々は、今回の研究会を通じXMLには多くのメリットがあり「XMLの可能性は無限である」ことを学んだ。4章の「XMLの有効利用」でも述べたように、ビジネス情報交換のための主要な媒体として今後、企業間電子商取引（B2B）などさまざまなシステムに積極的に使用されることが予想される。このような次世代技術であるXMLをどのように業務の中で生かせるかが、我々の今後の課題である。

また、今後次々と仕様が固まっていくと思われるXMLの動向に注目して行きたい。

### - 参考文献 -

1. まるごと図解 最新XMLがわかる  
技術評論社 池田実 / 小野寺尚希 著
2. これから始める人のXMLガイド  
日経BP社（日経ソフトウェア、日経オープンシステム）
3. XML MAGAZINE VOL.3  
翔泳社

### - 参考サイト -

1. アットマークアイティ  
<http://www.atmarkit.co.jp/>
2. 日経ソフトウェア（日経BP社）  
<http://software.nikkeibp.co.jp/software/index.html>
3. Prgraming Page  
[http://www16.freeweb.ne.jp/computer/mas\\_prod/](http://www16.freeweb.ne.jp/computer/mas_prod/)
4. Try!XML  
[http://user.shikoku.ne.jp/kyss/xml\\_xsl.htm](http://user.shikoku.ne.jp/kyss/xml_xsl.htm)
5. 日本XMLユーザーグループ WebPage  
<http://www.xml.gr.jp/>

## - 研究活動の感想 -

最後に今回の部会メンバー全員の感想を紹介する。

今回私がこの部会に参加したのは、上司から強引に勧められたことと「JavaもLinuxも知らないから新技術のXMLしかない」といった単純な理由からでした。入社以来メインフレーム系業務しか経験のない私にとって、初心者対象の入門書を読んでも説明している言葉の意味さえわからないという状況でした。研究会での討議と各種用語事典を片手に入門書を何度も読み返すことにより、ようやく「XMLの全体像」が見えてきたような気がします。今回このような機会を与えてくださったみなさんに感謝し、今後ともこの研究会を通じて得たことを業務に生かせるよう努力して行きたいと思います。

研究会のスタート時点では、XML自体の仕様が今よりも決まっていなかった事と自分自身の仕事がメインフレーム系である事もあり、本当にゼロからのスタートでした。グループのメンバーにも恵まれ久しぶりに学ぶことの楽しさを実感できた数ヶ月でした。実際の業務にて今後XMLを使う機会に恵まれるかは分かりませんが、今回の研究会に参加した事によって新しい技術に対する敷居は確実に低くなったと思います。

インターネットは見るもので作るものではなかった素人の私が、なぜかWebの最新技術「XML」をやってみたいと思いたちました。もともと興味はあったのでいいきっかけになったと思います。もちろん、参加資格が「入門者、または、興味のある方。」というのがきいていたのですが。まだまだ、作成できるレベルではないですが、基本の知識は得たかな？この研究会が始まりになるよう、今後も勉強していきたいと思えます。最後にメンバーの皆様、皆様のおかげで楽しく勉強できたことに感謝します。

XMLというものは見かけることがあっても、それがいったい何なのかということ全く知らない状態で、今回の勉強会に参加しました。私はこのXML部会に途中から参加したのですが、知らない用語が飛び交う中、最初の頃は、かなりとまどっていました。(意見を出そうにも、内容が理解できない...という苦しい部分もありました。)ですが、内容的には興味深い部分が多かったので、調べるのは大変でしたが、ためになったと思います。また、資料集めに関して言えば、ネット検索というもののありがたさを身にしみながら、「この検索をXML化したらどうなるんだろう？便利なのかな？」と考えたりもしました。今後の普及に期待！です。最後に、XMLに取り組む時間が少なく、調査不足などで皆様にご迷惑をかけた点もありましたが、楽しく会に参加できてよかったと思います。約半年と短い間でしたが、ありがとうございました。

この部会に参加したのは、部会に参加する以前に、HTMLを使用したシステムの構築に携わった事で、Web技術について興味を持ち初めたという事が理由でした。と

はいえXMLについては何も知らず、初めて聞く言葉ばかりで、私をはじめ、メンバー全員がまったくの手探り状態取り組んでいったという感じであったため、とても大変ではありましたが、その反面、わからないなりに意見を交わしていく事が楽しく感じることもあり、とても有意義に過ごす事ができました。この部会に参加させて頂き、本当にありがとうございました。

今回のXMLということで最先端の技術を勉強、研究、今後の動き等の考察を勉強できた時間は、有意義でした。また、いろんな人との交流、意見等を交わし、自分の見聞が広がったことは大きな収穫でした。ただ、やはりもう少し深い研究、勉強を行うには、少し環境が足りませんでした。今後は、環境面を充実させて研究会を行っていくことが必要ではないかと感じました。最後に本研究会のみなさんにご迷惑をかけたこととお詫びいたします。ありがとうございました。

XML研究会を通して、XML関連の技術が非常に多く存在するのを知りました。また、勉強するにつれ、XMLが情報共有のフォーマットになりうることを知ったとき、XMLがなぜ注目されているのか少しわかった気がします。これからXMLは更に普及していくのではないかと思います。XML研究会を通して得た知識が役に立つように、今後もXMLを勉強していきたいと思います。XML研究会に参加させて頂き、本当にありがとうございました。

今回OISA研究会に参加し、XMLという新しい技術に触れることができ大変よかったですと思います。しかし、活動途中で転勤ということで最後まで研究会に参加できなかったことは大変残念です。今後この経験がどこで活かせるかわかりませんが、常に新しい技術に挑戦するという姿勢を維持していきたいと思います。そういう意味でもOISA研究会への参加は有意義だったと思います。メンバーの皆様、大変お疲れ様でした。

それまで、あまり意識する事の無かった、"XML"という言葉であるが、この半年間は興味を持って、接する事が出来た。その結果、大手ポータルサイトを始め、多くの場面ですでにXMLが実際に使われはじめているという事が分かってきた。今後は、どのように実際の業務に活用できるかが、課題であると思う。