

3. XMLの概要と応用

池田 実 (東芝アドバンスシステム (株))

はじめに

現在インターネットが計り知れない恩恵をもたらしていることは疑いのない事実である。が、現在のWWWでは、提供側のサーバから情報がいわば垂れながし的に提供されているに過ぎず、WWWをEDIやECなどビジネスへ応用する場合など不都合も少なくない。

現行のHTMLは、画面表示を指向したマークアップであり、商用のデータのような情報の構造化が要請される応用には向かない。また、クライアント側では画面の表示に変化をつける程度の情報加工しかできず、自動化やエージェント機能の実現などが困難である。

図-1は、現状のWWWの機能的構成を表している。図の左側はサーバサイドの機能で、サーバの後方にはデータベースやその他のアプリケーションが接続され得る。また、図の右側はWWWのクライアント側の機能を示していて、HTML (Hyper Text Markup Language) ファイルの参照を主とするコミュニケーションの世界を実現している。つまり図の左側は高度化された情報処理の世界につながっているのに対して、右側のクライアント側の世界はHTMLで記述された内容を画面に表示するだけの単純な機能しかない。したがって、配布された情報の内容を解釈したり、計算を行ったりする処理はすべて人間作業となっている。

企業などにおけるコンピュータシステムの一般的な命題として、ある種の自動化、もっと言えば人間活動の部分的な代行が期待されている。その実現には、ビジネスデータが扱えて、かつソフトウェアなどによる情報の加工や処理ができなくてはならない。

こうした要請は当初のWWWの目的からは逸脱したものかもしれない。しかし、すでにネットワークの応用が単純な情報交換や共有を越えて、EC (Electric Comers) やEDI (Electric Data Interchange) などワールドワイドの商取引などに進もうとしている。

そのような要請を受けている次世代のWWWは、以下の2点を実現しなければならない。

1. プレゼンテーション目的のHTMLから、より内容記述に適した構造化言語XMLの導入
 2. ネットワークを流れる構造化された情報にアクセスし、アプリケーションが要請する処理ができること
1. の要件を実現すると見られるのがこれから説明するHTMLの上位規格である構造化マークアップ言語XMLであり、2. の要件を満たすのがJavaやActiveXのようなプログラミング言語機能とXMLの記述内容間のインタフェースとなるDOM (Document Object Model) 規格である。

新しいWWWの世界

前述の要件が満たされたとき、どのような姿がもたらされるのか想像してみよう。図-2はその予想である。XMLによって取り引きデータのような商用に耐え

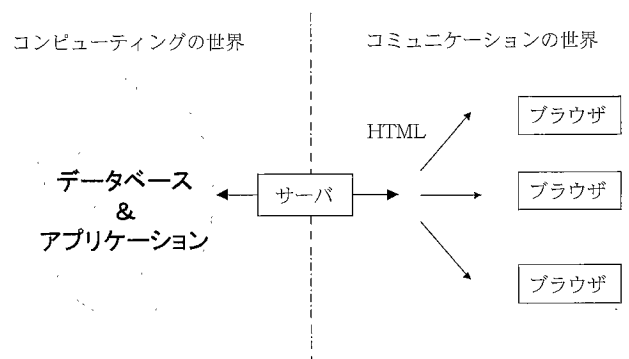


図-1 現在のWWWの機能構成

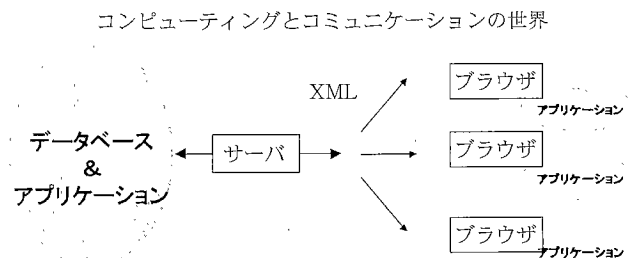


図-2 これからのWWWの機能構成

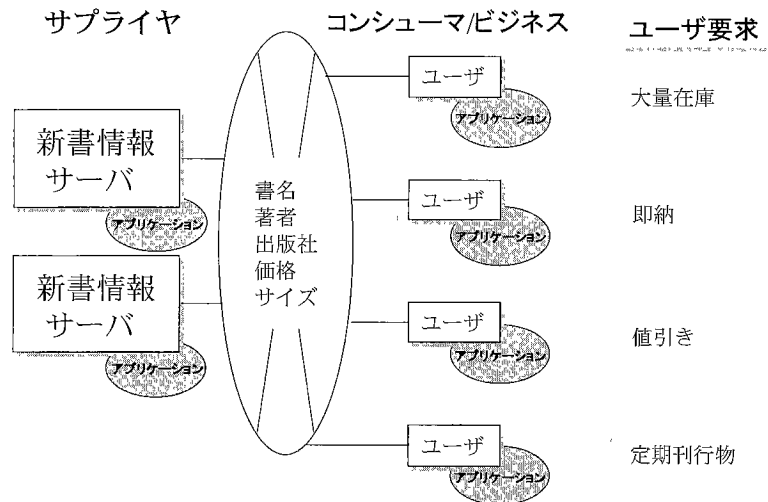


図-3 ビジネス/コンシューマ、ビジネス EDIシステム構成

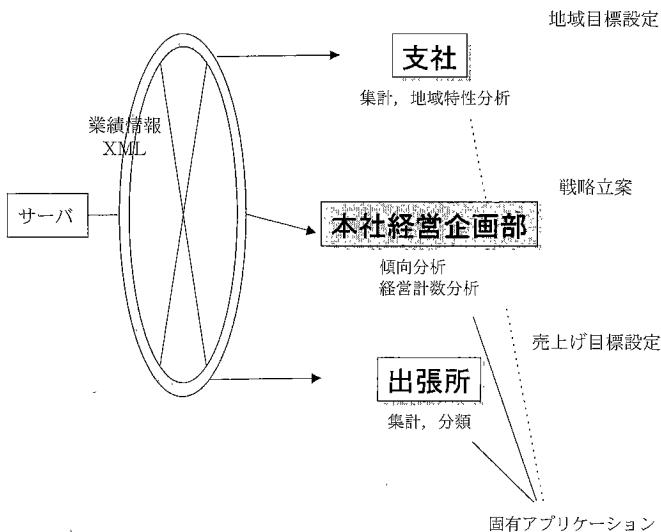


図-4 クライアント・アプリケーションの応用例

る構造化情報が配布できるようになる。それに加えクライアント側ではJavaなどのプログラミング言語が、XMLで記述された情報中の必要な要素に選択的にアクセスしたり、あるいは書き換えたりできる。こうした仕組みによってサーバ側とクライアント側が切れ目なくつながり、情報の流通と処理が一体化された世界が出現する。

これによってクライアントには、複数のWebサイトを巡航するようなある種の自動化や、発注などの取引を処理するためのアプリケーションロジックなどを持つことができる。

また別の観点から見ると、これまでクライアントでは高性能のパソコンを使っているながらHTMLの単純なブラウジングしかできなかったがこのモデルではサーバとクライアント間で負荷と処理の分散によってクライアント側のコンピュータ資源を有効活用すること

と、サーバ側の設備をリーズナブルな範囲に押さえることが両立している。

■ XMLの応用 ■

XML規約の根幹をなすパート1が、1998年2月10日にW3Cの正式な推奨案となった。しかし、実用的なシステム構成を構想するには環境が未熟と言わざるを得ない。なぜなら関連規格なども含め仕様自体も不確定な部分も少なくなく、システムの構築に必要なツールなどもとても十分とは言えないのが現状だからである。しかしながら、XMLの応用はきわめて現実的で今日的な課題であり、マイクロソフトを含む主要なインターネットのベンダがXMLに大きな力を注いでいる現状を見ると、事態は急速に改善されるものと予想される。

□ エージェント機能の実現

図-3は、WWWで新書情報を提供して受注を処理するモデルである。このモデルは、EDIの典型の1つであり、複数のサプライヤが各々自社のサーバを持ち販売情報を公開している。現在のWWWクライアントは、各々のサーバに自分でアクセスを繰り返して提供情報を集め、それらをメモなどで手元に記録しておき、最後に自分の希望が一番合ったサプライヤを見つけて発注を行う。つまり、発注先の選択までの過程は、ほとんどの作業が人間に委ねられている。

また、現行のWWWではすべてのアプリケーションはCGI (Common Gateway Interface) などを通じてサーバ側で処理される。アプリケーション的な処理がサーバに集中するだけでなく、情報の転送回数も多く、その過程で盗み見や改ざんを受ける危険性も増す。また、この方式では不特定多数が要求する多種多様な要求に応えるサービスをサーバ側が完備するのは

難しい。

こうした应用到XMLを用いると、クライアント・アプリケーションがユーザの要望を受け入れ、自立的(ローカル)に判断をくだしたり、処理を行うことができる。このような構成によって、ネットワークベースの集中/分散処理が両立するインテリジェントなEDIシステムの構築が可能となる。

図-3で、各ユーザごとに異なる要求、たとえばあるユーザは大量に購入したいので在庫が最も多いサプライヤに発注したい。また別のユーザは、緊急に必要なので、最も迅速に提供できるサプライヤに発注したいなどなどさまざまな要求があるはずである。

ここでクライアントアプリケーションがユーザの要求オプションに対して最適な解を探し出すようなエージェントソフトを用意すれば、ユーザが必要な要件を入力するだけで自動的に最適な解を見つけることができる。

ユーザの要求を受けたクライアント側のエージェントソフトは、関連する複数のWebサイトを巡航し、そこから条件に合う候補を探して一覧として画面に表示する。つまりユーザは、エージェントソフトが複数のサイトから情報を横断的に集めて作成した候補一覧から選択するだけで、最適な相手に発注することができる。

図-4も、XMLが構造化されたビジネスデータを配布できることと、クライアントごとに異なるアプリケーション・ソフトウェアを持てることを利用したアプリケーション例である。

□ 個別アプリケーション

図-4では、本社サイドから社内共通の情報として最新の業績情報を常時流している、これを受け取る地方の出張所では、所在する地域の情報だけを選択し、これに統計処理を施してその地域の今月の売上げ合計と目標の差を算出し、地域全体の売上げ動向を勘案した出張所の次月の売上げ目標を作成する。また、本社の経営企画部では、配布された全情報を統合しグローバル情報として再構成する。そのグローバル情報を元にシミュレーションを行い、企業全体に関する短期的な経営戦略を立案するための基礎資料を作成するような例である。

この例を別の視点から見てみる。図-5は、配布情報とそれを処理するクライアント側のアプリケーションソフトの関係である。左側の四角は提供情報を表し、縦方向は情報の種別である。この構成では、各クライアントは、それぞれが必要な情報に選択的にアクセスし、かつ独自の要求を処理するアプリケーションソフトウェアを備えている。この構成によってエンドユーザは社内全体に公開されている情報を利用しているにもかかわらず、それぞれが自分専用の情報とアプリケ

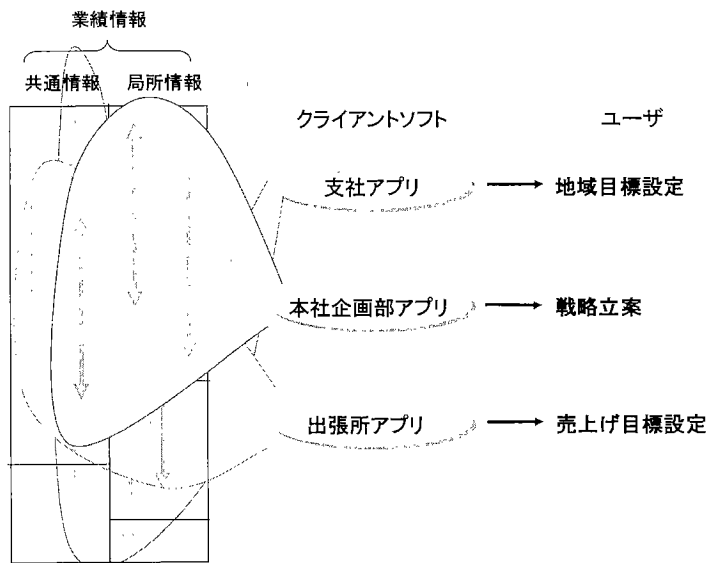


図-5 共通情報とクライアント固有アプリケーション



図-6 HTMLのタグ、要素の関係

```
<新書>
<書名>ジャパンふたたび</書名>
<著者>藤山 三郎</著者>
.
.
.
</新書>
```

図-7 XMLで同じ情報を表現する

ーションソフトを専有しているかのように利用できる。こうした仕組みが、サーバにおけるCGIやデータベースの支援がなくても、あるいは最小の支援で実現できるのである。

以上の2例は、XML応用のほんの入り口に過ぎない。今後、多くの有用な応用アイデアが出てくるに違いない。

■ HTMLの限界とXMLの解決 ■

これまでWWWで利用されてきたHTMLは、基本的にブラウザに対して表示を指示するためのマークアップである。つまり、アプリケーションで必要となるような“情報が持つ意味”を明確に表す構造表現ができなかった。また、HTMLのマークアップはあらかじめ規格として決められており、データベースのスキ

表-1

新書 (1)	
書名	ジャパンふたたび
著者	藤山 三郎
出版社	XX経済出版
価格	5,900
サイズ	B5
新書 (2)	
書名	縄文の不思議
著者	大山 太郎
出版社	YY美術出版
価格	3,200
サイズ	A4

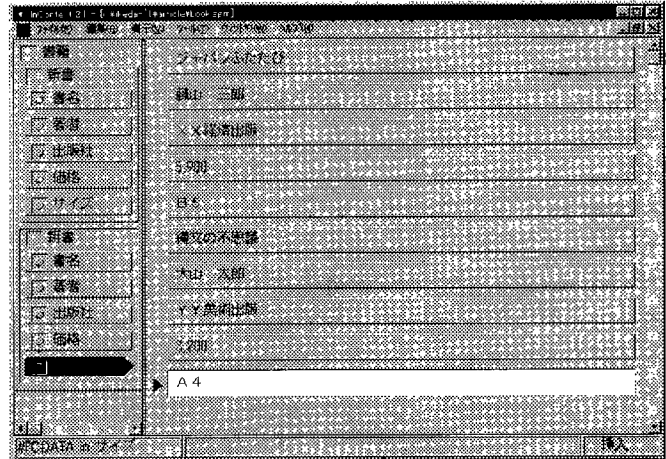


図-10 XMLドキュメントを入力する例

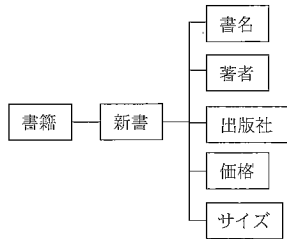
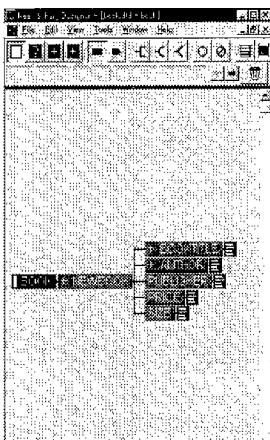


図-8 ツールで作成中のDTD (左) とその情報構造図 (右)

```

<?xml version="1.0" ?>
<!DOCTYPE BOOK [
<ELEMENT BOOK (NEWBOOK+)>
<ELEMENT NEWBOOK (BOOKTITLE+,AUTHOR+,PUBLISHER,PRICE,SIZE)+>
<ELEMENT BOOKTITLE (#PCDATA)>
<ELEMENT AUTHOR (#PCDATA)>
<ELEMENT PUBLISHER (#PCDATA)>
<ELEMENT PRICE (#PCDATA)>
<ELEMENT SIZE (#PCDATA)>
]
  
```

図-9 新書情報のDTD

マのようにユーザが自分のアプリケーションに合わせて設計する仕組みを欠いている。これも大きな制約である。

図-6のHTMLのタグは、要素内容“書名 ジャパンふたたび”を「p」の表示様式で画面に表示することを要求している。

表示を指示する情報としてはこれで十分だが、意味内容である‘書名’を情報として取り出そうとすると、どこからどこまでが書名なのか定かでない。これをXMLで表現すると、先の表形式と同様に明確に情報を表現できる。たとえば図-7のようになる。

■ XMLにはデータ定義がある ■

先に挙げた書籍データの表形式の表現を木構造の表現に変えたものが表-1である。

SGMLやXMLでは、ドキュメント内容を作成する前にこれから作成すべきドキュメント構造の定義を行わなければならない。その構造定義がDTDであって、DTDの記述にはDTD文法を知らなければならない。ここでは画面インタフェースを通して絵を描くようにしてDTDを設計できるNear&Far DesignerというSGML用の設計ツールの様子を図-8の左に、その結果できあがった構造図を図-8の右に示す。また、参考までに、このツールで作成したSGML用のDTDに手を加えて、XMLに準拠させた新書情報用のDTDを図-9に示しておく。なお、厳密には、要素の繰り返しなど、多少の省略をしてある。また、タグの名前はASCII文字にしてある。

このような構造定義はSGMLやXMLだけの特別な物でなく、データベースのスキーマ定義や表計算ソフトであらかじめ作成された表定義などもこれに類する。情報処理を目的としたシステムの一部と考えれば、むしろ当たり前のことだろう。

実際のインスタンスの作成について触れておこう。XMLのドキュメント内容はすべて文字で表される。したがって、文字が入力できるテキストエディタなどで編集可能である。しかし、厄介なDTDの規定に合わせなければならないので、神経が消耗するし、誤りも起き得る。そこで、SGMLやXMLにはDTDが規定するルールにそって入力を誘導するSGML/XMLエディタと呼ばれる編集ツールが存在する。図-10に示するのはInconTextというSGML用の編集ツールである。

画面の左側にDTDで定義されたドキュメント構造が表示され、各タグに対応して右側に要素内容の入力用のフィールドが配置される。この例からもSGMLやXMLの構造性が大変明確であることがわかる。

表-2

DTDの省略	
HTML	明示的DTD存在せず
XML	省略可能
	省略(適正) 付加(正当)
SGML	省略できない

この例では、<NEWBOOK>タグを「書籍」と表示するような置き換えを行っている。

• 適正なXML (Well Form)

SGMLとXMLではDTDが不可欠で、これによって構造を定義することを述べたが、実はXMLでは、一定の条件さえ満たせば、DTDなしで実体(インスタンス)だけでも運用できる仕様になっている。これを適正(Well Form)と呼ぶ。WWWでの応用を考えれば簡便で、実用性を高めるだろう。

一方SGMLでは、DTDの省略はいかなる場合も許されない。

• 正当なXML (Valid)

DTDとXMLの実体(インスタンス)をペアで運用する場合を正当(Valid)と呼ぶ。この状況ではツールがDTDとインスタンスの整合性をチェックする。

表-2にHTML, XMLおよびSGMLのDTDに関する省略の可否をまとめる。

■ XMLと関連規格 ■

XMLはHTMLに比べ、大きく拡張されている点がある。その1つは、ハイパーテキストリンクである。HTMLでは、点から点あるいは点からファイルへの単純なリンク機能しかなかったが、XMLではHyTime規格中(ISO/IEC 10744で規定された、ハイパーテキストと空間や時間の表現が可能なマルチメディアを指向したマークアップ言語)のいくつかのリンク仕様を取り込んでいる。

□ XMLのリンク (XLL)

XMLには多様なハイパーテキスト仕様が用意されている。ここではXLLの文法的な説明は抜きにXMLで利用できるハイパーテキスト形式についてその概要に触れることにする。XMLでは、リンク機能に大幅な拡張がなされている。

XMLでは、複数のリンク先を持つことができる。リンクを選択する操作は、リストからいずれかの分岐先を選択するような実装がなされると考えられている。

XLLの目新しい概念として、アウトラインリンク(Outline link)がある。これまでの常識として、リンクは起点と終点からなるものとされ、起点には何らかのリンク情報が埋め込まれ、それをクリックしたり

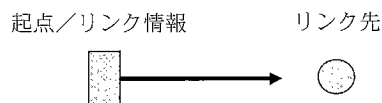


図-11 これまでのリンク(インラインリンク)



図-12 アウトラインリンク(複数リンク)

するとリンク先に移動するようなアクションを起こした。このアウトラインリンクは、リンクの起点とそれを保持する場所を分離する構造になっている。図-11を見ていただきたい、これがこれまでのリンクのリンクで、起点=リンク情報であり、起点にリンク先を表す情報が埋め込まれていた。この方式のリンクをXMLではインラインリンクと呼ぶ。

それに対してXMLのアウトラインリンクは、リンクの起点とリンク先を保持する場所を分けてしまった。図-12で真ん中の四角はリンク先を保持している場所であるが、起点ではない。起点は2つのリンク先になる。図では矢印が一方方向になっているが、双方向となるのが自然だろう。そのあたりの仕様は明確ではない。

このアウトラインリンクの良い点は、左右のドキュメントがすでに公開されている第三者の物であるような場合、新たに作成した真ん中のドキュメントが、左右をリンクすることによって、仮想的(あるいは間接的)に丸で表された左右のドキュメント間のリンクができることである。

つまり左右端のオリジナルドキュメントには一切手を加えなくても、四角で表されたアウトラインリンクを3番目のドキュメント内に設ければ、左右のドキュメント間のリンクが成立する。たとえば、公開されている日本語辞書とヘブライ語辞書の該当項目を第三者がリンクすると、日本語/ヘブライ語の対訳辞書が作成できる(原理的には)というわけである。

さらに、HTMLには存在しない概念として機能を持ったリンクがある。たとえば、「XX報告.XMLの“2番目の報告者”の“課題”へリンク」や「報告者名が“山田”の“理由”へリンク」などのような、記述構造や記述内容に基づく検索的なリンクも可能である。

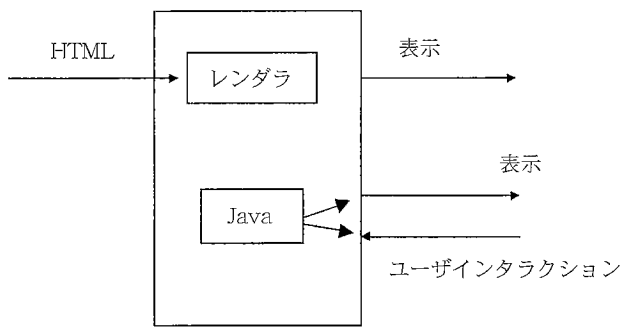


図-13 現在のJavaとHTMLの関係

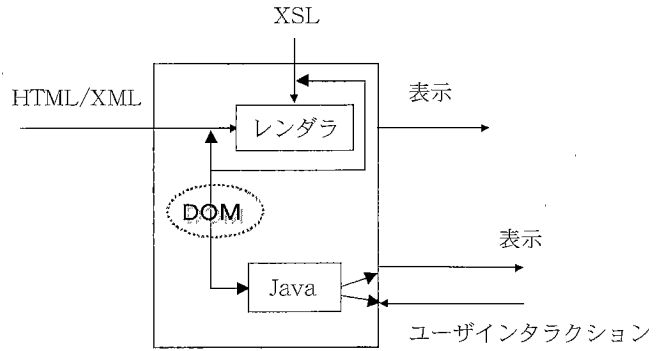


図-14 DOMインターフェースでHTML/XMLとJavaが連携

□ XSL

XSLはeXtensible Style Languageを意味し、XMLの内容に表示様式を与えるものである。HTMLでは各タグごとにその表示形式が決まっているが、XMLではユーザが自由にタグのセットを定義できるため、各要素に対する表示様式をブラウザがあらかじめ決めることはできない。そのためXMLでは表示方法を指示する仕組みが別途に必要となる。それがXSLである。

表示の様式を与えなければ表示できないのは面倒である反面、情報内容とその表示方式を分離することによって、これまでにない高度な応用が可能となるのである。

これまでのHTMLのブラウザではユーザは表示に関して何も指定しなくてよいかわり、基本的にはあらかじめブラウザが設定した固定的な表示しかできない。

自動的にタグ名で表示が決まるHTMLに対して、XMLではブラウザがXSLに記述された表示指定に従い表示を行う。つまり、XMLではXSLの指定がなければ、整った表示はできない。その代わりに、XSLによってユーザが自分の意志を反映させた表示を自由に設定でき、さらに必要であれば表示様式を動的に変化させるような高度化も可能である。

たとえば、同一のXML記述内容でありながら、状況が緊急を要する場合は、表示文字のポイントサイズをあげて大きく表示したり、赤字で注意を喚起したりする応用もある。あるいは、アクセス権限の低い人にはセキュリティレベルが高い内容を隠す（表示しない）ようなことも可能である。

また、1組のXMLに対して複数のXSLの記述を用意しておき、場合に応じて適用するXSLを切り替えることもできる。このようにXSLとXMLが独立していることにより、表示の内容と様式を動的に変化させることができるのである。

■ XMLとプログラミング ■

□ これまでのプログラミング

現在、WWW上におけるプログラミング言語の代表としてJavaが挙げられる。これまで実現されているアプリケーションは、ユーザのアクションを受け、それに応じて描画したりするものなどが主で、EDIなどで期待されるHTMLやXMLの内容を読み出ししたり、ドキュメント中のタブに設定されたアトリビュートの値を評価したりすることはできなかった。

次は、現状のビューアにおけるHTMLの表示機能（レンダラ）とJavaの関係である。HTMLで表された情報内容とJavaが提供するプログラミング機能が有機的に結びついていない。現在のHTMLは、プログラムの記述を含んだり、あるいは起動にかかわってはいるものの、HTMLの記述内容をJavaなどに引き渡すことができない。

図-14が、今後想定されるクライアント側の論理的な機能構成である。図中のDOMは、Javaなどのプログラム機能がHTMLやXMLにアクセスするためのインターフェースの規格であり、W3C内で検討が進んでいる。このDOMが実装された製品がユーザに提供されて初めて、これまでに述べたようなクライアントアプリケーションが実現できる。

DOMは、このようにJava（プログラム）とXML（データ）を結び付けるインターフェースの規格であり、今後のWWWにとって大変重要な規格である。

■ 最後に ■

規格の制定とは別に、激烈な競争下にあるインターネットの世界ではXMLとて例外ではありえず、マイクロソフト、サン、ネットスケープあるいはデータベースベンダや新規参入企業などを巻き込んで新しい競争が始まっている。その様子からは、意外に早くXMLの実用が始まりそうな気配がうかがえる。

(平成10年3月13日受付)

訂 正

本誌第39巻6号（1998年6月号）の中で次の個所に誤りがありました。お詫びして訂正いたします。

会誌編集委員会

誤

正

- | | |
|---|--|
| (1) p.505, 右段, 上から20行目: DVD (Digital Video Disk) | DVD *特定用語の略語ではない |
| (2) p.515, 左段, 下から5行目: EDI (Electric Data Interchange) | EDI (Electronic Data Interchange) |
| (3) p.515, 左段, 下から6行目: EC (Electric Comers) | EC (Electronic Commerce) |
| (4) p.515, 左段, 下から22行目: HTML (Hyper Text Markup Language) | HTML (HyperText Markup Language) *HyperTextは 語 |