

総 括

「SGMLは情報システムの基盤技術」の観点から¹

石塚 英弘²

図書館情報大学図書館情報学部

1. はじめに

SGML(Standard Generalized Markup Language)[1][2]は分野や業種を越えて使われる技術であり、情報システムの基盤技術[3]である。本稿はその観点から、なぜSGMLが有効なのかについて、新高度情報システム：CALISにおいて重要な情報共有を支援するからであることを述べる。次いで、SGMLの趣旨、HTMLとの違い、SGMLを導入した場合の実例などについて総括的に解説する。

2. 情報システム基盤技術としてのSGML

以前から欧米ではSGMLは分野や業種を越えて使われる技術であって、政府・公共機関、産業界を問わず広い範囲で使われるようになっていた。また、米国国防総省の文書電子化プロジェクトとしてスタートしたCALISでもSGMLが採用されていたが、そのCALISが民生化[4][5]され、インターネット上のグローバルかつ異業種・異分野に渡る情報共有を特徴とする新高度情報システムとして生まれ変わった。その結果、SGMLはこの新しい情報システムの基盤技術として位置づけられている。

一方、我が国ではSGMLは特定分野、たとえば印刷に限定されるとの認識が長らく続いていたが、SGMLが民生版CALISの基盤技術の一つであることが知られたこともあって認識が進み、最近では様々な分野や業種で使われるようになってきた。実際、十分な認識の下にSGMLを導入し、着実に情報システムを変革している企業が存在する。これは欧米特にアメリカの企業との取り引き、あるいは情報交換ないし仕事の分担をしている企業によく見られる。この状況を観ると、SGMLが情報システムの基盤技術であるとの認識が広まる時期も近いと感じる。

その一方で、未だSGMLの趣旨を認識しきれない状況も垣間見られる。たとえば、「今はSGMLが流行だから、一応付き合っておくか。やったことが無いでは不味だろう」との及び腰の態度の企業も見られる。この現象はたんに「認識が足りない」で済む問題ではなく、その背景に日本の企業や社会にとって苦手とする概念、即ち、他の組織との情報共有、が含まれているからではなかろうか。ネットワーク上で互いに必要な情報を共有し、分担した仕事を実行する、これこそが民生版CALIS(新高度情報システム)の姿である。そして、その情報共有を支える技術がSGMLである。その点、意識改革が必要だろう。

¹本稿は著者が情報処理, 37巻3号(1996.3)と、システム制御情報学会の講習会資料(1997.2)に書いた内容に加筆したものである。

²Hidehiro Ishizuka, Faculty of Library and Information Science, University of Library and Information Science

3. 情報共有を支える技術：SGML

日本人の発想は「縦割り」が基本だと言われる。この発想で、グローバルかつ異業種・異分野に渡る情報共有を特徴とする新高度情報システムに対応できるだろうか？ たとえば、縦割り組織はお役所だけでなく、大企業にも見られる。日本では学問体系まで、理学、工学、医学、薬学、農学、法学、経済学、文学と縦割りである。縦割りの向こうは別世界と見て「互いに知らない、話が通じない、各自勝手にやる、全体のことはどうなってもよい」では新高度情報システムへの対応は困難だろう。

それでは対策は何か。発想の転換、すなわち、「縦割り」から「横断」、そのためには、仲間内の情報交換から異業種・異分野の人たちとの情報交換・情報共有へといった転換が必要であろう。そして、その発想の転換を支え、情報交換を実現する技術の一つがSGMLである。それは文書DBの技術だからである。ここで文書とは、テキストだけでなく、図表ほか必要なものを含んだものをいう。

なぜなら、異業種・異分野では単純なデータを交換してもそれが何を表すかが分からない。データを補足する説明用文書が必要になるからである。たとえば、同じ自動車会社、あるいは系列会社間であれば、「部品番号12345の部品の在庫はいくつか?」、「300個あり」で済むし、この要求ならば関係型DBを共有すればよい。しかし、自動車会社が複数の会社カーナビの引き合いを出した場合はどうだろう。カーナビの性能を知るには関係型DBだけでは不十分で、製品カタログといった感じの補足情報が必要になろう。その情報はテキストと画像から構成されるから、DBになっていないと対応できない。さらに、自動車会社とコンピュータ会社との間で、自動車会社が「ガソリン化器のコンピュータ制御」を要求した場合はどうだろう。ガソリン化器の構造と機能、ガソリンと空気の混合比率等々の情報を含めた要求仕様書が必要になろう。これは正に図表を含む文書DBである。

説明用文書のDBを、必要になった時点で一々最初から作っていたのでは急場に間に合わない。元となる文書DBを通常の業務の中で日頃から作っておき、そのDBの中から必要な項目を検索して取り出し、その目的に合ったDBを構成する方が効率的である。関係型DBに載るデータについては、DBの中から必要に応じて抜き出せるようになっているが、文書データについてはワープロのデータしかないことが我が国では少なくないようである。この状況をSGML方式の導入により変革していく必要がある。

勿論、情報共有とセキュリティは表裏一体の事柄である。共有して差し支えない情報の内容は相手により異なる。従来のDBでも相手によりアクセスできるデータは異なっていたから、文書DBについてもそれを可能にする必要がある。SGMLはこの課題にも対応できる。項目単位だけでなく、文章中の文言レベルでアクセス権を変えるための印を付けることができるからである。具体的には、秘密にしたい文言の開始個所に開始タグ(tag)を、終了個所に終了タグを入れることにより他の部分と区別する。そして、表示ソフトウェアの方で、そのタグを識別し、アクセス権を持たない利用者に対しては該当個所をスキップして詰めて表示する。その結果、利用者は秘密部分がかったことも知らずに読み終ることになる。

4. SGMLの趣旨

4.1. 文書構造の表現

SGML[6][7][8][9][10][11][12]の趣旨は文書の構造を表現することである。検索機能や表示機能などは敢えて持たず、「餅屋は餅屋」、専門のシステムに委ねている。構造を厳密に表現しておけば、それから各種の検索表示システムあるいは印刷システムの内部フォーマットにプログラムで自動変換できるからである。また、構造が記述されていれば、構造を手掛かりとした検索、たとえば、章または節のタイトルに「WWW」という単語が含まれている章または節の中で、「マルチメディア」という単語が含まれている段落を検索することも可能となる。こうすれば、たんに「マルチメディア」で全文検索するよりもノイズの少ない検索ができる。この手法は、操作マニュアルや仕様書の内容を更新する際に該当箇所を探す方法としても有効である。

さらに、文書構造を記述しておくことにより、異なるハード・ソフト環境でも使える文書DBが構築できるし、使用していたハードやソフトが時代遅れとなり、別のハード・ソフトに乗り換えることになっても元のデータベースは生き続けられることになる。ハード・ソフト環境に依らないから、情報交換・情報共有に適しており、また保存形式としても適しているのである。

SGMLでは、文書の構造を表現するために、まず文書構造を定義する。この文書構造定義をSGMLでは、DTD (Document Type Definition、文書型定義) という。DTDでは文書の構成要素の名前と互いの構造上の関係、すなわち、章、節といった階層関係、また図や表の参照といった参照関係、を定義する。そして、DTDに従って実際のタイトル、著者名、テキスト、図、表、式等々を、generalized markup (汎用マーク付け) という記法を使って書く。

なお、SGML自体はDTDを書くための言語である。もしデータベースの用語に例えれば、DTDはスキーマに、SGMLはDDLに対応する。実際の文書のテキストはDTDに従ってマーク付けすることになるから、DTDは文書の記述の仕方を書いたものといえる。また、SGMLはISO規格で、JIS規格でもあるが、これはDTDを書くための言語の文法を規格にしたのであって、文書構造そのものを規格化したのではない。

ご承知のように、文書によりその構造は異なる。たとえば、単行本は章節で構成されているのに対し、雑誌は各種の記事の集まりとして構成されている。そこで、単行本用のDTD、雑誌用のDTDなど、文書の種類に応じたDTDを用意する。また、構成要素も様々である。論文の場合は抄録が付くが、単行本には付かない。図、表、写真、参考文献などは多くの文書に共通に存在するが、数式、定理の証明、化学構造式、文章の引用などは扱う内容によって存在したり、しなかったりする。もっとも、これまでに様々な機関で色々なDTDが作られてきているので、それを使えば自分で作る必要はない。また、既に存在するDTDを一部書き換えてもよいし、最初から書いてもよい。なお、前にCALSはSGMLを採用したと書いたが、それは、CALSの文書データベースはCALS用のDTDに従って書かれていることを示す。

以前、「SGML方式」という言い方をしたが、その理由は、用語“SGML”の意味はDTDを書く

くための言語に限定し、DTDに基づいたDBやシステムについては「SGML方式」として区別するためであった。

なお、markupはmark upから作られた造語である。mark up（マーク付け）とは、編集者がゴチ、10ボなどの活字の指定やセンタリングや図表の位置などの割付指示を原稿に赤字で入れることである。この指示を電子的に与えることをmarkupと一語で言う。マーク付けには固有と汎用がある。固有マーク付けの典型例は電算写植の印刷用制御コードで、これは使用するハード・ソフト固有である。一方、汎用の方は文書の体裁ではなく、構造を記述するmarkupで、文書の構成要素の始点と終点をマークする。こうすれば、マーク付け自体はハード・ソフトに依らなくなるから汎用となる。

4.2. 柔軟性

SGMLは柔軟性が高い。この性質はSGMLを他の規格と一緒に使うときに便利である。これもCALSにSGMLが採用された理由の一つだろう。

SGML方式ではテキスト・データだけでなく、図や写真なども扱えるようにしている。図や写真はビットパターン化したファイルとし、それを外部データ（SGMLの用語では外部エンティティという）として付ける。そして、それを文書のテキストから参照する。画像の形式にはGIF (Graphics Interchange Format), TIFF (Tag Image File Format), PS (PostScript) 等々いろいろな種類があるが、SGML方式ではDTDで定義すればどれでも使える。

文字データについてもSGML方式は柔軟である。たとえば、数式や表はSGMLの書法でも書けるが、DTDで定義しておけば数式や表の部分のみLaTeXで書くこともできる。また、外字もDTDで名前を定義することによって、テキスト中では&名前;の形式で書ける。名前はASCII文字で書くから、データの交換流通も安心してできる。

映像や音声についても外部エンティティとして扱うことができる。MPEG、QuickTimeなどの形式は問わない。ただし、これはファイルとして扱うので、その中身を分けて扱うわけではない。そこで、映像・音声の中身を扱うためのISO、JIS規格として、SGMLの拡張であるIlyTime[13]が定められている。

4.3. SGML方式文書DBの構成

SGML宣言(SGML declaration)、DTD、文書のテキスト、それに図や写真をビットパターン化したファイル(外部エンティティ)の4つで構成される。

SGML宣言には、使用する文字セットや処理系に要求するSGMLの機能などを書く。文書の言語が、英語、日本語など何で書かれていても、文字セットの指定で対応できる。

DTDは文書を構成する要素(element)とその相互関係、すなわち階層関係と参照関係をELEMENT文、ATTLIST文、ENTITY文を使って定義する。図1に、単行本のDTDを例に挙げて説明する。

本は、タイトル、著者、目次などの前付け(front matter, 以下fmと略)、章・節・段落などで構成される本体(body, bdyと略)、索引、奥書などの後付け(back matter, bmと

<!ENTITY % floats "fig tabl note">	<monogr>
<!ELEMENT monogr - - (fm, bdy, bm?)	<fm>
+ (%floats;)>	<tit>情報システム基盤技術としてのSGML
<!ELEMENT fm - 0 (tit, sht?, aut, toc)>	<aut>石塚英弘
<!ELEMENT bdy - 0 (chp+)>	(中略)
<!ELEMENT bm - 0 (indx?, phls?, phdt?)>	<bdy>
<!ELEMENT chp - 0 (chptit, (sec+ p+)>	<chp><chptit>はじめに
<!ELEMENT sec - 0 (sectit, p+)>	<p>SGML... 解説する。
<!ELEMENT p - 0 (#PCDATA)+>	(中略)
(中略)	<chp><chptit>SGMLの趣旨
<!ELEMENT fig - - (nf?, figcap?, fighdy)>	<sec><sectit>文書構造の表現
<!ATTLIST fig id ID #IMPLIED	(中略)
file CDATA #IMPLIED	<sec><sectit>SGML方式文書DBの構成
type CDATA #IMPLIED	<p>SGML宣言... の4つで構成される。
(中略)>	(中略)
<!ELEMENT figref - 0 EMPTY>	<p>DTDは... 図1<figref refid=fig1>に....
<!ATTLIST figref refid IDREF #IMPLIED>	(以下大幅に略す)
(以下略)	</monogr>

図1 単行本のDTD(一部)

階層構造はELEMENT文を使って、図表などの参照関係はATTLIST文で定義する属性を使って表現する。

図2 SGML方式文書のテキストの例

図1のDTDに従って書いたSGML方式文書のテキスト。要素ごとに、その開始点を示すタグ<要素名>が付いている。

略)といった階層関係を持っている。図1の2行目で、element名"monogr"はfm, bdy, bmから構成され、出現順はfm, bdy, bmの順であることを示している。要素が順に出現するときはカンマで区切り、どちらかが現れるときは| (OR)を、両方が現れるが順は問わないときは& (AND)を使う。要素の出現頻度が、0または1のときは?で、1からN回のときは+で、0からN回のときは*で示す。たとえば、4行目は、本体は章の繰り返しで構成されることを、6行目は、章は章のタイトルと、複数の節または段落で構成されることを表現している。なお、#PCDATAは解析対象文字データという意味で、予約語である。

ELEMENT文は要素の始点・終点のマークが省略できるか否かも示す。"0"は省略可、"- "は否を示す。左が始点で、右が終点を示す。

図表や注などは本文からの参照関係を持つ。そこで、まず1行目と2行目で、図、表または注などを%floats;とし、monogrのどこにも現れうることを示す。そして、参照・被参照の関係をIDを使って表現する。IDは11行目や15行目のように属性リストの一つとして定義する。図の場合は、ファイル名は属性fileで、GIFほかの画像の形式は属性typeで示すことを定義すればよい(12、13行目参照)。

仮に「情報システム基盤技術としてのSGML」というタイトルの単行本があり、それが図1のDTDに従うとすれば、その文書のテキストは図2に示すようになる。

そして、図を参照する場合は、たとえば、図2の下から3行目に示すように本文中には<figref refid=fig1>と書く。この記述はfig1をIDとして図を参照することを示している。図そのものは別ファイルにあるが、IDを使って参照するわけである。なお、図1の下から3行目のELEMENT文で figref EMPTYとあるのは、文書テキスト中にはfigrefのテキストデータがないことを示していたのである。表や注の場合も同じようにIDを使って参照できる。

ところで、このようなく要素名>や</要素名>付きの複雑な文書テキストを書くのは面倒である。そこで、通常の文章を書く感覚で入力すれば、自動的にSGML方式の文書テキストに変換してくれるツールが開発されている。欧米だけでなく、最近では日本でもそれが普及してきたから、容易にSGML方式の文書テキストが作れるようになってきた。また、SGML方

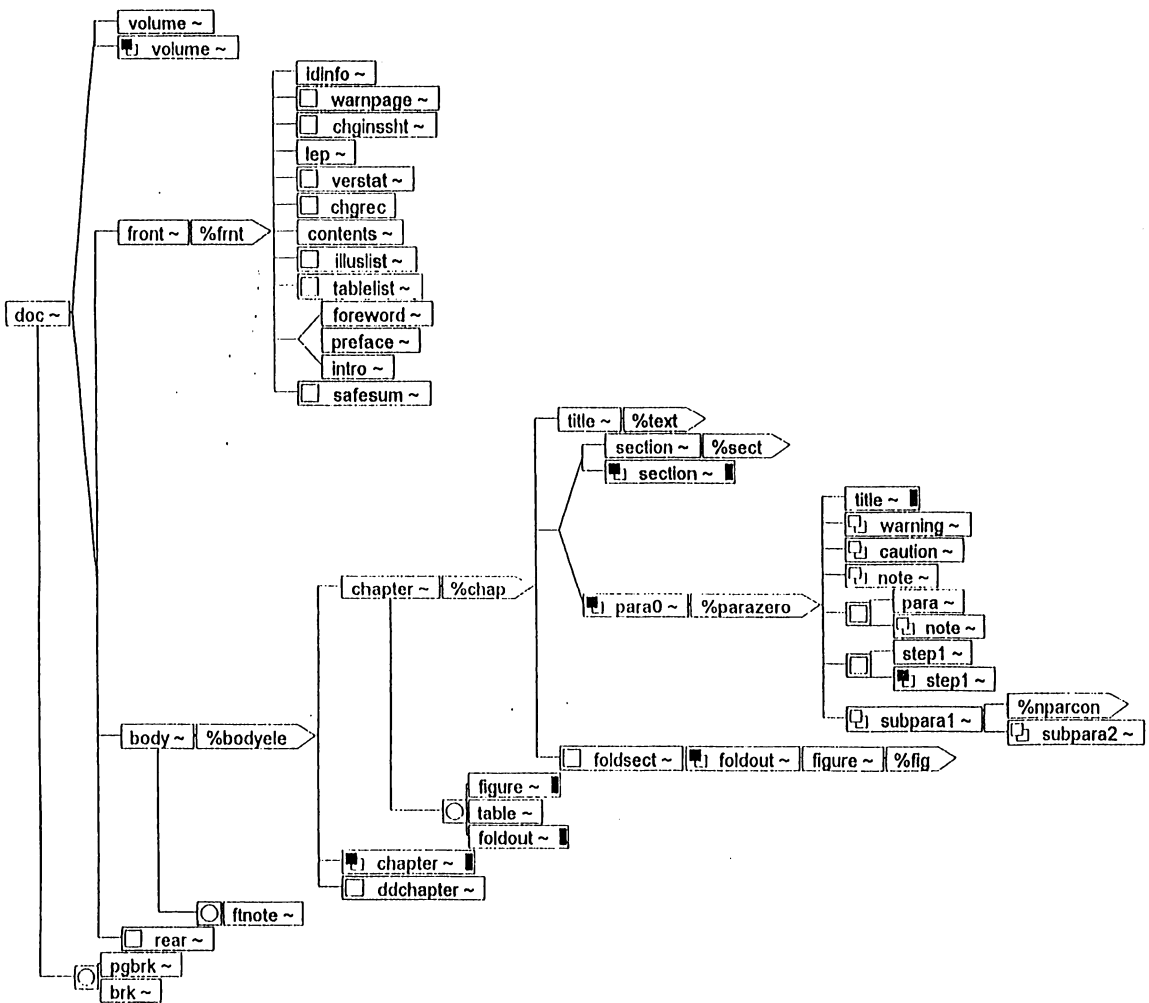


図3 米国CALSのDTD (M11.38784C) の木構造表示

式の文書テキストはSGML対応のDTPあるいはviewerを使えば綺麗にレイアウトされる[14]から、SGML方式の文書テキストはダンプ・リストのようなものと思っていればよい。

また、SGML方式文書DBにはDTDが付いているから、それを基に別の構造の文書DBにツールを使って変換することもできる。そのため、SGML方式文書DBは交換流通に適しているのである。

4.4. 米国CALISのDTD

前述のDTDは説明用の単純なものであったが、実際のDTDは対象とする文書の構造が複雑なためDTDもかなり複雑なものとなる。たとえば、米国CALISのDTD (MIL Standard 38784c) は複雑な階層構造を持つため、DTDの文を観るだけでは構造が分かり難い。これに対応して、これを木構造で表示し、またその構造をGUIで変更できるDTD設計支援ソフトウェアが開発され、市販されている。カナダのMicrostar Software 社の“Near & Far”はその一例である。そこで、これを使って、米国CALISのDTD (MIL Standard 38784c) を表示した(図3参照)。要素の名前が書かれた箱が木構造のノードになっている。箱をマウスでクリックすると、その下部の構成要素が表示される。一番右に見える“subpara2”の右にもさらに下部の構成要素が存在するが、図としては大きくなりすぎるため図3では表示していない。

5. HTMLの趣旨とSGMLとの相違点

HTMLはHyperText Markup Languageの略だが、これは文字どおりHTMLの趣旨を表している。すなわち、WWWというハイパーテキスト・システム専用のDTDに従ったマーク付け記法によるデータ入力書式がHTMLである。HTMLは、WWWのハイパーテキスト機能を実現するための書式になっているが、他の検索表示システムには対応しない。また、HTMLはWWWで表示するための記法であって、SGMLのように文書の構造を汎用的に表現するものではない。HTMLのDTDをNEAR & FARを使って木構造で表現すると図4のようになり、章、節といった階層構造はないことが分かる。また、見出し項目(たとえば、H1, H2ほか)はあるが、著者という項目はない。

SGMLならば、DTDを自分で書くことができるが、HTMLの場合は書いたり修正することはできない。HTMLはWWWの機能に結びついた記法であり、もしもHTMLの記法を変えたとするとWWWの機能を変えなければならないからである。

もっとも、SGML方式でDTDに従って書く文書テキストの記法はマーク付けであり、その点はHTMLによく似ている。そのため、「HTMLとSGMLは同じようなものだ」と誤解する人や「HTMLはSGMLのサブセット」という誤解を呼ぶ言い方が出たのだろう。しかし、これまで説明したように、HTMLとSGMLは趣旨が異なるのだから、サブセットという言い方は適切でない。

WWWは確かに優れた検索表示ソフトウェアである。しかし、Mosaic, Netscape, HotJava[15]と少しずつ異なるソフトがあるし、もっと優れたソフトが次々に出てくる可能性が高い。表示に重点を置いたHTML形式よりは、構造を表現したSGML方式でデータを保持した方がソフトの変化に対応しやすいだろう。

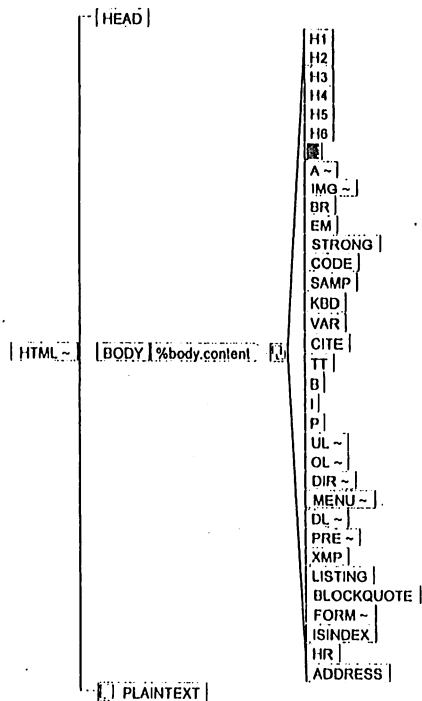


図4 HTMLのDTDの木構造表示

なお、ブラウザで見た目はWWWでも、WWWサーバの方はHTMLではなくSGML方式で持っているものがある。たとえば、データは検索エンジン・サーバ中のSGML対応検索エンジン、たとえばOpenTextの中に格納しておき、検索はそのエンジンで実行し、検索結果を受け取ったWWWサーバがHTMLに変換してWWWブラウザに送るものがある。あるいは、WWWサーバからSGML文書DBのままでWWWブラウザに送ってもらい、それに接続したSGML文書DB用viewer：SoftQuad社のPanorama[16]で表示する方法もある。Panoramaは検索機能も持っている。

6. SGMLを導入した場合の実際

SGMLを導入した場合、発生する仕事は何か。それは、1)全社的な方針の確立、2)文書構造の分析とDTDの設計、3)SGML方式に対応した情報システムの開発、4)そのシステムの運用、である。

6.1. 組織としての方針の確立

SGMLを導入して、組織を越えた情報共有を目的とする新高度情報システムを作るのであるから、全社的な方針の確立は必須である。また、新システムの利用者もこの変革によるメリットを自らの問題として認識し、積極的に取り組む必要がある。なぜなら、これまではDBとはなっておらず、そのため規制が緩やかだった文書に関する仕事も情報システムの中に明確に位置づけられ、そのことによって日々の仕事の仕方が変わるからである。

6.2. 文書構造の分析とDTDの設計

扱う文書の構造を分析し、その結果に基づいてDTDを設計する。もっとも、従来のDB分析設計者も文書構造を分析した経験はあまりないから、実例をよく集め、注意深く分析する必要がある。また、その分野で既にDTDが存在することもあるから、その場合はそれを

十分検討すべきである。検討の結果、同じDTDを採用することもある。そのDTDを変更する場合も安易に変えるのではなく、十分に検討して、データの互換性を配慮して決める必要がある。その分野で既存のDTDがある場合は、そのDTDに従ったDBとデータ交換する可能性が高いからである。既存のDTDで公開されているものとしては、たとえば、前述のCALSのDTD、米国出版協会が作ったDTDがISOになったISO 12083などがある。

DTDを検討・設計する場合は、前述のソフトウェア：Near & Far が有効である。これを使って設計したDTDであっても、SGMLの文法に従っているかを調べるソフトウェアであるSGML parser、たとえば、Mark-Itあるいはsgmlsで文法チェックする必要がある。また、このDTDが現実を反映したものかどうかを実際の文書データでチェックする必要があるが、その際もSGML parserを使う。SGML parserは文書データがDTDに従っているかをチェックし、エラーが無ければフルにSGMLのタグが付いた形式に変換する。

6.3. SGML方式に対応した情報システムの開発

SGML方式に対応するには、データ入力部分、DB検索・管理エンジン、表示部分を用意する必要がある。この3つを持った文書管理システムとしては、InterleafやDynaTextがある。また、文書管理機能だけでなく、情報システムとしての機能を加えたものとしては、たとえば、Information Managerがある。また、入出力部分はWWWを使い、DB検索・管理エンジンにSGML方式のエンジン：OpenText、あるいはオブジェクト指向DBMS：Oracleを使う方法もある。

DTDに従ったデータ入力のためのソフトウェアはSGMLエディタと呼ばれ、たとえば、Xerox社のXSoft InContextやSoftQuad社のAuthor/Editorなどがある。また、ワープロソフトでSGML方式の文書テキストを出力する機能を持ったもの、たとえば、MicrosoftのWordやジャストシステムの一太郎などがあるし、LaTeXで書かれたテキストを変換プログラムによりSGML方式の文書テキストに変換することもできる。ただ、SGMLエディタに比べるとSGMLの文法チェック機能が弱いので、SGMLパーサによるチェックを経る必要がある。

6.4. システムの運用とフィードバック、外部とのデータ交換など

従来の情報システムの運用と同様、運用の過程で問題が出た場合はフィードバックし、問題を解決する必要がある。たとえば、DTDの不備が見つかった場合は慎重な検討を経てDTDを改良することになる。

外部とのデータ交換では、異なるDTDを持つ場合は要素間の対応表により変換の仕方を定義し、自動変換することができる。たとえば、日立製作所のIntegraは変換ツールを持っている。勿論、変換ソフトを自作してもよい。

SGML方式の文書データを送ることも可能である。たとえば、前述のPanoramaは、WWWを通してSGML方式の文書データ（HTMLデータではない）を受け取り、それを表示することができるし、検索機能も持っている。

7. 終りに代えて

CALSの浸透に伴い、SGMLは日本でも常識化するだろう[17]。すなわち、SGMLツールが普及し、SGMLの面倒な点は意識せずに使えるようになるだろう。事実、最近のツールの普及は著しい。

また、見方を変えれば日本ではあまり注目されていない側面があることに気づく。電子出版は印刷業界や出版業界の話に留まらない。WWWに見られるように、ネットワーク上の電子出版[14]は情報共有の新しい形態である。「出版」という日本語の言葉で観ると違和感があるかもしれないが、元の英語publishing（公開すること）で観れば分かりやすい。欧米には古今東西のテキストを電子化するプロジェクト：Text Encoding Initiative [18]があり、それでは当初からSGMLが採用されている。また、関連してネットワーク上に仮想的な図書館：電子図書館[19]が作られつつある。

このようにして近い将来、情報システムがSGML方式文書DBを扱うことが一般化し、利用者はSGMLを意識せずに自然に使うことになるだろう。そして、その時には従来とは大きく異なる21世紀の新高度情報システムが実現しているだろう。

文献

- [1] ISO 8879, Information processing - Text and office systems - Standard Generalized Markup Language (SGML).
- [2] JIS X 4151, 文書記述言語SGML, 日本規格協会.
- [3] 石塚英弘, 根岸正光: 情報処理最前線: 情報システム基盤技術としてのSGML 一文書データベースからWWWそしてCALSまで一, 情報処理, 37, 3, pp. 207-212 (1996. 3)
- [4] 後藤龍男: CALS: 21世紀における企業情報システムの国際標準確立と企業統合に向けて, 情報処理, 36, 1, pp. 1-7 (1995).
- [5] 水田浩: CALSの可能性, 256p., 生産性出版, (1995).
- [6] 吉岡誠編著: SGMLのススメ, 167p., オーム社, 1993.
- [7] 根岸正光・石塚英弘共編: SGMLの活用, 168p., オーム社, 1994.
- [8] (株)日本ユニテックSGMLサロン編著: はじめてのSGML, 285p, 技術評論社, 1995
- [9] 吉岡誠編著: SGMLを使いこなす, 185p., オーム社, 1996.
- [10] Eric van Herwijnen著, SGML 懇談会実用化WG監訳: 実践SGML, 日本規格協会, 1992.
- [11] Martin Bryan著, 福島誠訳: SGML入門, アスキー出版局, 1991.
- [12] C.F.Goldfarb: The SGML Handbook, 663p., Oxford University Press, 1990.
- [13] 小町祐史: マルチメディア/ハイパメディア情報交換の標準化動向, 情報処理, 35, 7, pp. 632-641 (1994); 小町祐史: HyTimeの規定, 文献[7]のpp. 90-99, 1994.
- [14] 石塚英弘: 電子出版 その概念と技術, 電子情報通信学会誌, Vol.78, No.9, pp. 891-898(1995);
- [15] J.Gosling, H.McGilton: The Java Language Environment A White Paper, 65p. Sun Microsystems, May 1995 (Sun Microsystemsによる翻訳あり).
- [16] 機能紹介が(www.softquad.com/products/panorama/panor-fe.htm)にある. また, デモ版が(www.oclc.org/5046/oclc/research/panorama/panorama.html)からダウンロードできる.
- [17] 根岸正光: SGML普及への展望, 文献[7]のpp. 144-164, 1994
- [18] 長瀬真理: テキスト・データベースとTEI, 文献[7]のpp. 117-141, 1994.
- [19] 長尾真: 電子図書館, 岩波書店, 1994.