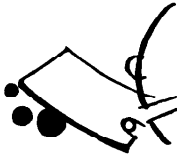


報 告

パネル討論会

卓上出版のネットワーク化†



パネリスト

石田 晴久¹⁾, 小町 祐史²⁾, 春田 勝彦³⁾
 森 正樹⁴⁾, 中島 聡⁵⁾
 司会 田丸 啓吉⁶⁾

司会(田丸) 本日は「卓上出版のネットワーク化」というテーマでパネルディスカッションを開きます。卓上出版あるいは卓上出版のネットワーク化ということは技術的な問題もまだたくさんありますし、そのほかに標準化の話とか、用途の問題や経済性の問題、さらにもっと大きくいいますと出版文化というところまで関係するような問題を含んでいるわけです。限られた時間ですけれども、大学およびメーカ、通信、出版、ソフトウェアなどいろんな分野の企業の専門の方においでいただいておりますので、いろいろ面白いご意見を伺えるのではないかと考えています。最初に石田先生からお願いします。



Postscript 言語の適用

石田 ネットワークの上に文書情報を流すときに、私のところでは今のところ Postscript という形式を使っています。Postscript というのはいわゆるページ記述言語と呼ばれる一種のプログラム言語ですけれども、Adobe という会社が 1984 年 8 月に出したんです。そのころキャノンから LBP-8 という小型の卓上型レーザプリンタが出たわけですが、1985 年 1 月にアップル社が、キャノンのこのエンジンを使いまして、レーザライタというのを出しました。その中に Postscript という言語を入れて、Postscript で書かれたテキストをレーザライタに送ると、



非常にきれいな印刷ができるということを示したわけです。

それを使いまして、1985 年 7 月にアルダスという会社がページメーカと呼ぶ一種のレイアウトソフトウェアを出しました。その中で Postscript を使うということをやしまして、同時にこれの宣伝文句として、デスクトップ・パブリッシングという言葉を作ったんですね。これが定着してこういうことをやるのを DTP と呼ぶようになった。日本語で直訳すれば、卓上出版ということになります。この Postscript ですけれども、これは IBM が一応採用したということ、それから日本では NEC から Postscript 内蔵のレーザ・プリンタが出るというような情勢になってきましたので、日本もある程度普及が見込めるわけです。

それで、私どもで前からネットワークの上でマルチメディア文書、つまり文字情報だけでなく、図形とか写真が入ったものを送るには、どうしたらいいか検討していたんですけれども、どうもあまりいい知恵がなかった。それであるとき、この Postscript のことを調べて、これは面白そうだから当面これでいこうということにしたわけです。

それで、今いましたページメーカで作る文書はもちろん Postscript で出てくるわけです。それから Unix では troff とか Tex とかという一種の英文の清書プログラムがありますけれども、それから出てきたテキストを Postscript 型式に変換するコンバータを通しますと、全部これが Postscript 型式になる。

それから Unix では Plot といって、図形情報を表現するソフトがあるんですけれども、その出力も変換すれば、Postscript になる。また Unix では S という有名な統計解析パッケージがありまして、それを使いますと統計解析した結果をいろんなグラフにすることができるようなんですけれども、それも Postscript で出す機能が AT&T から標準的に出されていますので、そ

† 日時 昭和 63 年 11 月 15 日 (火) 15:30~17:00 の「マイクロコンピュータとワークステーションによる卓上出版とネットワークシンポジウム」

場所 機械振興会館大ホール (地下 2 階)

1) 東大, 2) 松下電送, 3) NTT, 4) オーム社, 5) マイクロソフト, 6) 京大

ういうものを使うと結果全部 Postscript になる。したがってソフトはみな違うんですけれども、出てくるデータはみな同じ型式になる。だからここで一つの標準化ができるわけです。

それで、これをディスプレイでみたいときには、このハードウェアにしろ、ディバイスドライバだけ書いてやれば、みられる。それからレーザ・プリンタに出して、ハードコピーをとりたいというときに、このレーザ・プリンタが、Postscript をサポートしていれば出せるわけです。さらに私どもの場合には、自分たちでディバイスドライバを作っていますので、プリンタは必ずしも Postscript をサポートしていなくても、ソフト的にディバイスドライバで出すことをやっています。このことは、ネットワークをとおして、こういう情報を流してやればそれを受け取って、自分で見ることができることになる。

それで、これから日本語ワープロみたいなものの出力が Postscript 型式になるとよい。そうすると、それはワープロから見ると一種のデータなんです。一方レーザ・プリンタのコントロールとかディスプレイのコントロールから見ると、これは一種のプログラムです。ですから Postscript はプログラムでありデータであると、そんな恰好をしているわけです。

それから、この Postscript が大変面白いのは、写真のような画像も簡単に表現できる。それで写真のパターンを一つ取り込むと、あとは拡大、縮小はまったく自由ですし、それから回転も自由にできる。

このコードの効率をどう測るかなんですけれども、まず一般の英文のテキストを書いたときに、どのくらいふくらむかという、私がこの Postscript について英語の論文を書いた経験でいいますと、本来の英文は 14 キロバイトぐらいだったわけなんですけれども、Postscript に変換しますと、それが 34 キロバイト、つまり 2.3 倍ぐらいにふくらむんです。それから画像の場合ですけれども、1137 バイトのデータでもってスヌーピーの画像が構成されると、そんな感じになります。

ところで、英文のテキストを送った場合に、バイト数がなぜ 2.3 倍になってしまうかというと、英語の文章を Postscript のコードに変換させますと、単語一つ一つがカッコの中に入っていて、しかも前後に 4 桁の数字と終りに S という文字がついて 1 行になっちゃうわけです。要するに、プロポーショナル・スペーシングのために、Postscript では単語と単語の間のス

ペースというのが一定じゃありませんで、そのために単語と単語の間をどれだけあけるかという情報が、必要なわけです。それを表す情報が、各単語の前に 4 桁で入っているわけです。

また先の S というのは、そこでスペースをあけなさいという一種のマクロです。そういうのが入ってきますので、どうしてももとのテキストに比べて平均的に 2 倍ちょっとに長くなる。逆にこういうふうにしておけばいろいろフォントを変えたり、サイズを変えたりができます。

以上は英文の話ですけれども、これから日本でやることを期待されているのは漢字混り文です。その場合に Postscript では普通のドットベースの字ではなくて、輪郭線フォントというのを使いたいわけです。輪郭線フォントというのは、たとえば「愛」という字を表すときに、輪郭線をたどって、しかも曲がり具合にいくつかのポイントをさだめておいて、点と点の間をどのように結ぶかということ、一種の方程式にしてデータとしていれておく。それを表示するほうでは座標軸の大きさの縮尺を変えれば、字の大きさが自在になりますので、こういうものをネットワークでも送りたいわけです。

ただ、私のところにはまだ日本語の Postscript が使えるプリンタというのがありませんので、日本語の文章を送ったときに、さっきいった効率がどのくらいかという測定がまだできておりませんけれども、これからそういうプリンタが手に入ったら少し考えてみたいと思っています。

それで日本の場合は、さっきいった単語と単語の間にスペースを入れるということがありませんので、そのスペース数に関する情報は、おそらく送らなくていいだろう。そうすると主として 2 バイトのコードがずっと並んでいて、ところどころに制御情報ということですから、それほど効率は落ちないんじゃないかという気がしているんです。

最後にネットワークということで、私どもで行っていることというとネットワークの中でファイル共有をすることです。今私の Unix のネットワークの中においてあるフォントは、まださっきいった輪郭線フォントにはなっていないで、大日本印刷のドットベースのフォントです。サイズの違うドットのパターンが入っているものですから、全体の分量がかなり大きくて 130 メガバイトぐらいになるわけです。こうなるとそれをネットワークの中の各ワークステーションに全部

入れておくことはできませんので、どこか1カ所に置いておいて、それをネットワーク・ファイル・システムというのを使って、みんなで共同利用する。そういうことをやっています。

それからカナ漢字変換のための辞書ですね。カナ漢字変換のパッケージとしては、京都大学で作られた Wnn というのがよく使われているんですけども、そのための辞書を1カ所に置いてあります。というのは、みんなでこれからボキャブラリをふやそうということがあるものですから、個々のワークステーションに入れておくんじゃなくて、どこかに置いておく。それでもネットワークのスピードが速ければそれほどオーバーヘッドにならない感じなのです。共用できる。

以上きわめて現実的な話ですけれども、卓上出版とネットワークの絡みというと、私も現在はこのことをやっているということを報告しておきたいと思っています。

司会 それでは続きまして小町さんをお願いします。

パッケージメディアによる電子出版と標準化

小町 私は ISO の SC 18、つまりテキスト・アンド・オフィスシステムの国際標準化をやっておりますグループのメンバという位置付けでお話させていた



だきます。ネットワーク化といいますと狭義には通信回線を使ったり伝送を指しますが、もう少し広い意味にとりまして、パッケージメディア、つまりリムーバブルなストレージメディアを使って、電子化文書を配布、流通するために必要な標準化にフォーカスを当てて、お話ししたいと思います。

それでここで配布・流通の対象になっている電子化文書が従来のハードコピー文書に比べてどういう点で優れているかを、まとめてみました。まず論理構造、レイアウト構造、それからコンテンツという形ではっきり分離できるということが、文書を作る上でも再編集する上でも扱いを非常に便利にします。もともと出版分野ではステップごとに専門家がいて、それぞれ別の作業をやっておったんですが、ワープロができてからはそれらが一つにまとまってしまっています。ところが卓上出版になって、出版作業がより一層分業化できる環境に近づきつつあります。電子化された文

書では、管理、保管が容易になります。検索、登録がしやすくなるのは当然ですが、さらにハイパertextを使いますとオブジェクト間のリンク付けであるとか、リンクをたどっていくということ、これはハードコピーでは絶対不可能ですが、それが可能になります。また、閲覧のログギング、これは課金につながりますが、こういったものが容易になります。

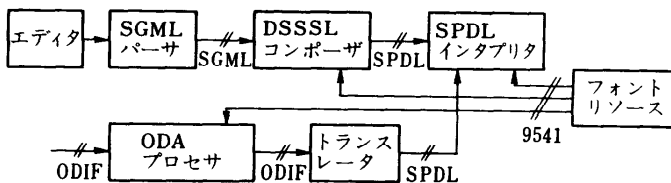
たとえば百科辞典を購入しましても端から端まで読むということはないんです。一部分しか読まないにもかかわらず、全部の金を払わされるというのは、読者にとっては都合が悪い。むしろ見たところだけの金を払いたいというのが望むところなんです。社会的にできるかどうか分かりませんが、技術的にはこの要求は満たされるようになります。

配布、流通ということ。SC 18 ではどういうふうに考えているかが、ごく粗っぽく SC 18 WG 1 のモデルに示されています。ドキュメントを作るブロックと、これをトランスフェル、ストレージするブロックとがユーザ・システム・インタフェースと関係しています。それらを標準化することと、いずれにも関係のあるセキュリティを規定することを SC 18 の課題としています。

このドキュメント・ストレージとトランスファの部分と同じ SC 18 のドキュメントがもう少し詳しく示しています。つまり ODA 文書があると、それをインタチェンジするために、ODIF、あるいは SDIF というフォーマットが決まっています。

午前中講演がありました DSSSL とか SPDL という言語による文書記述については、図-1 に関係を示します。つまり SGML パーサをとおってできた SGML 文書はインタチェンジの対象データですが、それをさらに DSSSL コンポーザにかけた結果、SPDL で記述された文書が出てまいります。これも交換の対象になるでしょう。さきほど石田先生の話にありました Postscript もこれと同様のページ記述言語ですが、ISO では SPDL という標準言語を開発しており、特に標準フォントリソースとの受渡しを規定しています。

DSSSL コンポーザとか SPDL インタプリタがフォントリソースをもってくるところも、インタチェンジの対象になります。フォントリソースについては 9541 という規格を決めようとしています。これをどう使うかに関しては、フォントサービスという新しい標準化対象が検討されております。



SGML (Standard Generalized Markup Language): 汎用マークアップ言語, DSSSL (Document Style Semantics and Specification Language): 文書スタイル記述言語, SPDL (Standard Page Description Language): 標準ページ記述言語, ODA (Office Document Architecture): 事務文書体系, ODIF (Office Document Interchange Format): 事務文書体系データ交換形式

図-1 文書作成モデルにおける受渡し対象文書情報

これらのインタチェンジの対象は通信系を介して送ってもよいし、もう一つの重要なディストリビューションの方法であるパッケージ・メディアで送ることもできます。通信系の場合には、オープンシステム間のインタチェンジのためのセブンレア・モデルができ上がっておりまして、レアごとに標準化されています。ストレージメディアについては明確なレamodelは定義されていませんが、同様の考えかたは必要でして、それがオープンな環境で交換されようとする、当然すべて標準化しなくてはならないということになります。ここでは大きく分けて三つのレベルの標準化が必要であろうと思います。

物理フォーマットというのは、たとえばディスクとかカートリッジの機械的特性、セクタの数や容量を規定したり光ディスクの場合ですとサーボ方式を決めたりするレベルです。それが与える物理的な記憶機能を使って、アプリケーションが使いやすいデータユニットを作り上げるのがボリュームファイル・フォーマットでして、それも当然、標準化の対象になります。具体的にはファイル、ディレクトリ、ボリューム、こういった記憶単位の標準化です。これが実際に文書などを納め込むユニットになるわけです。それを使うアプリケーションとしては、文書の構造と内容を、たとえば ODA あるいは SGML のように、決めておく必要があります。また光ディスクのように大量の文書をストレージできるメディアについては、検索データ構造のようなものをアプリケーションレベルで決めてやらなければなりません。

現在どういった標準化がなされているかといいますと、まず物理レベルにつきましては、フロッピーディスクがすでに標準化されています。CD-ROM の規格は、フィリップス、ソニーが決めたものを ECMA がほぼ追認しておりますが、まだ ISO にはなっており

ません。追記形の光ディスクに関しては、9171 という規格がほぼできています。書き換え形は現在検討中であり、いずれも SC 23 という組織によって行われています。

その上のボリュームファイルレベルでは、フロッピーに関して 9293 が決められています。9660 という CD-ROM のボリュームファイル・フォーマットはハイシェラグループが規定したものを、ISO が一部修正してオーソ

ライズしたものです。追記形あるいは書き換え形光ディスクについては、まだアプルーブされたものはありません。いくつかの案が出ているという状態になっています。

アプリケーションレベルにつきましては、ODA (8613) あるいは SGML という規格ができ上がっておりまして、それをどういうふうにフロッピーに載せ込むか、あるいは 9660 のメディアの上に載せ込むかという規格を、今検討しています。ODA 文書を 9293 の上に書き込む方法を規定しているのが DP 10033 でして、あまり部厚いものじゃないんですが、文書とファイルの関係、マルチボリュームの設定法が指定されています。この方法をいろんなパッケージ・メディアに拡張しようとしたとき、メディアごとに DP 10033 と同様の規格を作っていくのは適当でない、なんとかもうちょっと一般性のある方法をとれないか、ということが今 WG 3 で問題になっていまして、新しいワークアイテムの提案がされております。

次に光ディスクのような大容量パッケージ・メディアで問題になるのは、その内容の検索です。たとえば ODA のドキュメントですと、ドキュメントの一番最初にドキュメント・プロファイルというのがありまして、その中にキーワードなどの検索のための属性が含まれています。しかし、これが一つのドキュメント内に入り込んでいるため、光ディスクを受け取ったところが、その内容を全部なめてすべての文書の検索属性をシステムの中に持ち込む必要があります。そこから与えられたキーワードをその中でサーチして、希望する文書のアドレスを求めるといったことになります。ところがこの全内容の走査をするには、大体数時間から十数時間がかかってしまいます。これではクイックなインタチェンジができません。

そこで、検索属性の集合を、どこか 1 カ所に、ある

いは非常に少ない数のファイルの上に載せ込むということが必要になります。当然それもインタチェンジの対象になりまして、そのデータ構造も標準化することが要求されます。これについては、SC 18 の WG 1 という組織が先月行った国際会議の中で、これを新しいワークアイテムにしようとする日本の提案が一応おりました。それで ISO としての今後の検討課題になると思います。

こんな具合にパッケージ系というのは通信系に比べて少し遅れてスタートしていますが、将来の電子出版の環境を作るためには不可欠だということで、ISO で標準化への活動が積極的に行われております。

司会 それでは続きまして、春田さんをお願いします。

分散環境の文書印刷応用

春田 オフィスの中には種々のサービスとかアプリケーションがあるわけですが、これらのサービスを実現するファイルサーバ、印刷サーバなどを、みんないっしょに通信で結んでしまって、利用者がワークステーションから制御するような分散環境で印刷サーバを考えましたときに、次に述べるような4つの問題点が考えられます。



1 点目は文書構造の多様性の問題であります。代表的な文書の例に、一つはメール、二つ目は現在分散オフィスアプリケーションモデル (DOAH) で検討中の文書ファイリング (DFR) で考えられておりますファイルの2種類があります。

メールは郵便とまったく同じようなアナログで封筒部と内容がありまして、その内容は基本的にはデジタル信号であれば何でもよろしいということです。現在考えておりますのは、通常の IA No. 5 のテキストとそれからデジタル信号、これはデジタル音声でもミュージックでも何でもよろしいんですけれども、デジタル化した信号、G3 ファクシミリ、G4 ファクシミリ、テレテックス、ビデオテックス、それから暗号化された情報、さらにメールの中のメールという本体部の中にさらに封筒と内容を含むメッセージが入るというネスト構造、さらにややこしいのは、勝手にお互いが約束しているものなら何でも許されることも認められています。そういうものをメールを使って送ることができます。

次は、分散オフィスの中で考えている DFR に格納されるドキュメントの構造なんですけれども、ODA の文書でもよろしいし、メールでもよろしいのです。さらにグループといって複数の文書をまとめたもの。あとは単純なリファレンスということで、他の文書を参照してもよろしいし、さらに面白いことにどこかの図書館とか外部参照が可能です。こういう文書構造をもつ文書を印刷する場合、たとえばあるグループの印刷をする場合にはグループを指定しますと、このグループに属する内容を印刷する。リファレンスがあればリファレンスの中のポイントをたぐって中身をみる、そういう処理を考えなくてはいけないことになります。

第2点目は、印刷の対象とするドキュメントの属性についての取扱いが問題となります。たとえばメールの場合はメール本体部にある見出し (heading) といわれているところに、メールの属性を入れているわけで、たとえば標題とか書いた人が全部ここに入っているわけです。一方ファイリングでいいますと、メールの見出しに相当するものにはファイルを作った人とか作った日などの属性が入っているわけです。したがってメールの見出しの中の属性と、ファイリングの中の属性とは同じ属性でも中味が違っているわけです。それらの属性を単純に印刷した場合、読む人が誤解を生じないかという問題があります。

それから3番目の利用者のほうですけれども、今われわれ一番よく考えていますのは、たとえばメールボックスで受けたメールについて、保存印刷のためファイルに入れるわけです。その場合にファイルに入れるときの文書の作者名というのは基本的にはメールの宛先の名前と同じものになるわけです。しかしメールを受け取るときの名前とファイルのアクセスの名前が違う身というのは非常に使いにくい。逆にファイルの中をメールで送る場合にはファイルでいいますと、作者の名前ですね、それをメールでいいますと、発信者の名前に変換するとか、そういう名前という観点から全体を統一処理する必要があります。ユーザからみまして一つの OA 空間、すなわち OA 環境といえますけれどもそういうものを実現する必要があります。

第4番目に問題になるのはセキュリティの問題だろうと思います。通常セキュリティのある文書というのは、通信で送るかどうかという話はいろいろな議論があると思いますけれども、万一送った場合に、通常は電子化されたあるプロテクションで守るわけです。た

例えば、どうしても印刷したい場合どうするか。普通われわれは印刷して後でⒼという判こをポンと押しているわけですが、そういうことを押すだけでいいのか。オフィスの中での全体的な統一的なセキュリティが必要と思われる。そういった問題は印刷サーバを実際に実現する上で、かなり大きなウェートを占めると考えられます。

最後に、現在考えられております印刷サーバの機能ですけれども、おおまかにいって3種類ございまして、ジョブ印刷の依頼と、問合せ管理と運用管理です。印刷依頼というのは業者から単に中身をみずにこういうものを印刷してくれという要求だけをやる。さらに結果の報告も行う。2番目は問合せの管理ということで印刷しているジョブがどうなっているかという問合せをしたり、もうやめたいという取消しの話。3番目は利用者管理ということで、印刷サーバに対してこういう利用者を登録したいとか、リソースの管理とかそういう管理運用という観点であります。

司会 それでは続きまして出版社の立場から森さんをお願いします。

電子出版と出版現場

森 電子出版というと、実は1981年に、日本語ワードプロセッサの本を、当時は16ドットの今からみれば非常に貧弱な品質のワープロの出力を版下にして、大胆にも本を作ったということがあるわけです。作って見たら世の中で一番早く日本語ワープロによるカメラ・レディ・コピーの本を作ったということで、NHKの朝のテレビの話題になりました。内部では、ああいう質のものを本にしているのかという議論がおきました。そのころ、電子出版の多少研究めいたことをやっております、当時ワードプロセッサで入力をして、電子写植機にかけますと、とにかく校正刷りという印画紙で出てくるわけです。印画紙というのは、コストが高くて赤字が入ったりすると、また全部が印画で出てくる。これではとてもコストがかかんで、この方式ではだめだということで、ワープロのゲラ刷りを中間に入れる方式が考え出され、それで発行したのが、その半年後だったと思うんですが、こうしてコンピュータ組版方式の試行錯誤がはじまりました。

その当時から、いわゆる電子出版ということに対する定義を考えました。出版の概念をもう少し広くとら

え出版社というのは、紙・刷り・製本で本を作ることだけでなく、世の中にある潜在的な情報価値がある意図の下に組織立てて読者のみなさんが手に入れやすいような状態に加工していく立場ではなからうか。そういうとらえ方をしまして、第一電子出版、第二電子出版、第三電子出版を定義づけたわけです。

第一電子出版というのは、通常の紙・刷り・製本の書籍形態、雑誌形態をもっているものです。プロセスが電子化されて作られていく。一番進んだものではオンラインで原稿を送って、それがもっとも進んだ自動版下作成機にかけられて、本になっていく。あるいは雑誌になっていく。実際そういった作り方がされているものも、現実にあります。

第二電子出版は、いわゆるニューメディア出版というもので、フロッピーディスクあるいはCD-ROMなどを媒体にしたもので、オーディオテープとかビデオテープ出版物もこの部類に入るかと思えます。

第三電子出版は、オンラインによって知識情報を直接サービスする形態というふうなとらえ方をしました。

かなり早い段階でこういう整理をして取り組んできたことが、その後の時代のいろいろな商品開発などの基礎研究に役立ってきたわけです。

出版の実際の世界で本ができていく流れでは、原稿用紙に著者の方が執筆されると、それが編集者の手元に送られる。そして編集者がそれに手を入れて印刷のほうへ廻る。じつはこの過程にヒューマンな問題がありまして、たぶんDTPが進んでも、その問題は解決されない。第一に正しい日本語で書かれているかということです。そこにかかなりの神経を編集者は注いでいる。「赤字を入れる」という言葉がありますが、「テニヲハ」から、文体や用字用語の統一、そういったようなことを編集者がやっているわけです。

DTPは企業内出版ということでオフィスに入ろうとしていますが、一般のオフィスには編集者がいない。一方出版社にはコンピュータを使えるエンジニアがいない。これは双方いないとこのシステムは成り立たないんじゃないかという問題がある。そういった点で印刷業界の方々の方が、かなり勉強されておまして、だいぶ進んでいるのではなからうかと思うわけです。

電子化された流れということになりますと、原稿自体がフロッピーディスクで入ってくる。そして最近レーザ・プリンタの質が大変上がっておりますから、



場合によっては、多少の加工を施してそのまま版下に
して、印刷屋さんへ送って製本というルートで流れる
ことが現実には相当数出てきております。これが第一電
子出版です。第二電子出版はここであまり深く触れる
必要はないと思いますけれども、いろんな媒体がある
と、それらの媒体に応じてどういった企画内容が向い
ているか吟味して、これからは媒体が選べるというふ
うに思います。

最後にこういったシステムを導入していく、あるい
は普及していくということになりますと、人の問題と
いうことが非常に大きいかと思えます。編集者が個人
的な感覚で受けつけられないというアレルギー現象がある。
キーボード・アレルギーも根強くあるんじゃないだろう
か。そういったようなことでもっと編集者インタフ
ェースという点もご考慮いただければ有難いと思ひ
ます。

司会 それではソフト会社の立場から中島さんにお
願います。

ネットワーク環境におけるドキュメント作成

中島 今回はソフトウェアを
作っている者の観点から話をし
てみたいと思います。OS を作
る場合にはアプリケーションプ
ログラムを組み合わせて最終的
にエンド・ユーザにどういう環境を提供したらいいか
ということを常に念頭において、行います。そういう
観点からいって、どういうことを念頭において、今
OS の設計をしているか、もしくはアプリケーション
を作っているかということを簡単にお話したいと思ひ
ます。



まずネットワークと DTP といったときに、企業内
で LAN で結ばれているパソコンというレベルで、企
業の使う人は決して出版部内の人に限らず、すべての
事務の作業をしている人がワークステーションに坐っ
て、ドキュメントの作成をしているオフィスの将来像
をまず念頭においています。

ここで重要な点はアプリケーションプログラムとい
うのは独立したソフトウェア会社から出てきたものを
組み合わせたいというのが目標だという点です。これ
は決してテクニカルな話ではありませんが、現状の
マーケットとしてそうなっていますから、それをサ
ポートするために、OS はまったく独自に、相手のこ
とを知らずに作ったアプリケーションをどう組み合わ

せていったらいいかということを考えていかなければ
いけないということです。

いろんなアプリケーション—たとえばグラフを作る
アプリケーションだとか、レイアウトするアプリケー
ション、それからデータベース・アプリケーション—
が OS で標準化されたインタフェースを使って、コミ
ュニケーションして、それでいかなる組み合わせでも
動かなければいけない。グラフはこのパッケージを使
いたいけれども、表計算はこちらを使うといった選択
肢まで与えてあげなければいけない。

たとえばウィンドシステムでいうカット・アンド・
ペーストは、表計算をするアプリケーションで作った
表をレイアウト・ソフトに貼り込むといったときに使
われます。しかし、実際にネットワーク環境になって
複数の人間が同一データにアクセスするということに
なると、これだけではいけません。たとえば A さん
が、B さんが作っている表を貼り込んだドキュメント
を作る場合、A さんも B さんも同時に作業をするわけ
ですね。この場合、A さんがレイアウトしているとき
に B さんの作っている表がどこまでリアルタイムに反
映されてくるかということが重要になります。これは
単なるカット・アンド・ペーストでは B さんが変更す
るたびに、ペーストしなければいけないということに
なります。これを解決するには、なんらかのデータの
リンクが必要です。もちろんパフォーマンスの問題が
ありますけれども、OS が解決していかなければなら
ない問題の一つと思っております。

次の例は、DTP です。私のいる会社でも、多くの
人がページメカを使ったりしてドキュメントを作成
しております。そして実際に仕上がりのいいものは作
れるようになっていきます。ただ決して仕事の効率は上
がっていないんです。それはページメカのソフトが
とても面白いという悪い欠点もあるんですけども、
その人は本来ドキュメントを書けばいい時間にレイア
ウトして楽しんでしまうという欠点がある。これは、
これから現実企業でコンピュータを仕事の能率を上
げるために取り入れていくためには、ぜひ考えていか
なければならないことです。そのときに、ひょっとし
たら WYSIWYG というものはもう切り離してしま
ったほうがいいかもしれないという議論も一つすべ
きだと思います。

つまり、たとえばオフィスだったら、ドキュメント
のフォーマットというのは会社で定められているわけ
ですね。そしたら、そのレイアウトのスタイルシート

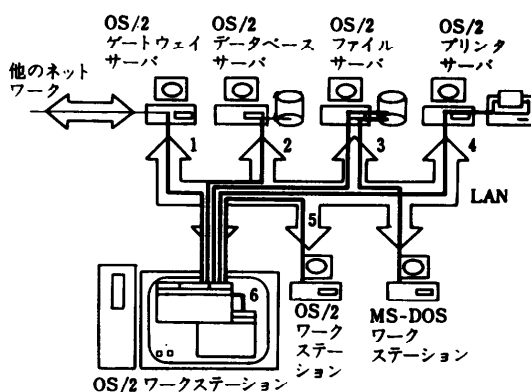
は会社の出版部門の人にやっていただく。その人がフォントのどれを選ぶとかいうことは全部する。

それでドキュメントを作る人は WYSIWYG でない環境でタイトルと名前と文章だけ打てば、きれいにフォーマットされた出力が出てくる、そういう形態のほうがはるかに仕事の効率を上げるという意味では役に立つんじゃないでしょうか。若干、天邪鬼ない方ですけども、これもぜひ考えていかなければならない方法であると思います。

レイアウトの話は、ネットワークが関係すると、さらに切り離しが重要になってきます。たとえば相手のプリンタがどのフォントをサポートしているかも分からない、どんな用紙に打つかも分からない、という環境でレイアウトしたり、修飾したドキュメントを送らなければいけない。その場合にはたぶんフォントはこれです、それから位置は何インチで、この位置におきますというインフォメーションを転送していたのでは、必ずしも相手のプリンタは打てないでしょう。将来的にいつか、全世界で全フォントが打てるようになってしまえばいいんですけども、現状としてはそうっていないでしょう。その場合には、ある意味で段落だけだとか見出しだとかという論理的な情報を送ったほうが、実際の用途としては役に立つでしょう。これは同一の企業内でも十分起こりうることだと思います。

もう一つは少し先になるかも知れませんが、一つの大きなドキュメントを複数の人間で扱う場合についてです。そのときに、たとえば段落単位もしくは章単位でそれぞれの人がドキュメントを作成するとか、相手が書いた文書に対してコメントをつけ加えて、ドキュメントをセーブしておくと、本来の執筆者がみると相手のコメントが入っているというような形態もたぶん考えなければいけないでしょう。これはかなり OS というより、アプリケーションが頑張らなければならない範囲の話ですけども、ファイルのロックだとかドキュメントロックという話が出てきますから、OS としても避けて通れない話だろうと思っています。

最後はデータベース・パブリッシングについてです。この言葉をだれが作ったか知りませんが、レイアウト情報とかフォント情報とかというものを一つの次元としたときに、データをもう一つの次元として二次元の出版をするということです。たとえば学校では、全生徒の成績表を一つの表で管理します。ですけれども各生徒に配るシートはその人の成績だけがに入った通信簿が先生のコメント付きで入ります。こういったドク



- 1, 2 ネームドバイブ 5 メールスロット
3, 4 リダイレクション 6 DDE

図-2 OS/2 ネットワーク環境におけるデスクトップパブリッシング

ュメント作成のときには、それぞれの生徒に対してレイアウトする必要は決してなくて、そのレイアウトはそのデータベースに対して直交した軸で行いたいわけです。このような形の出版形態をデータベース・パブリッシングと呼んで定義しています。たぶんこれは企業の仕事を効率化させるという意味ですぐにでも役に立つんじゃないかと思っています。

最後に、以上のような話を OS/2、および MSDOS を組み合わせた環境で、このように位置付けるということを表したのが図-2 です。細かくいうと OS がネームドバイブとかリダイレクションとか DDE とかいうローレベルのプロトコルで、アプリケーション間、ワークステーション間、もしくはワークステーションサーバ間の通信方法を提供しています。もっとも、まだこれだけでは不足で、こういったローレベルのレイヤの上でどういうドキュメントを、どういうフォーマットでという話をもっと議論を積み重ねていかなければいけないと思っています。

司会 それでは一通り講師の方のお話を伺ったんですが、何か補足をなさる方いらっしゃいますか。

小町 一番最後に中島さんから、レイアウトイングが問題であるといわれました。これはまさにユーザの立場から、デスクトップパブリッシングあるいは電子出版環境というものを適切に表現していただいた言葉ではなかろうかと思います。つまり、SC 18 の WG 8 なんか考えている要求項目をそのままユーザから指摘されたことにほかならないと思ひまして、非常に意を強くしているんです。やはりレイアウトというのは著者の仕事とは別の次元の重要な作業じゃないか、そ

してそこにこそ、先ほど森さんがいわれたように、出版社が今後積極的にでていく場があるんじゃないかなという気がするんです。

電子出版の環境において、出版社はどのような位置付けになるか、テレマティック国際研究所で研究しているんですが、たとえば SGML のようなかっこうで、つまりレイアウトが一切ないような形でどどん著者がデータベースにアクセスし、読者はそれをすぐ検索するということになりますと、レイアウトのない非常に簡単な出版形態ができます。それはそれで非常に価値があると思うのですが、そういう論理構造だけの文書は読者にとって必ずしもみやすいものではありません。日本語がいかどうかという話がありましたけれども、そのセマンティック・チェックはおそらくプログラムでできるでしょう。でもレイアウト付けとなるとそれは非常に高度な人間の活動を必要とします。それこそ、たとえば DSSSL のような言語を使って出版社がやる作業として今後さらに重要になってくるのではないかなという気がします。

そこで次に問題になるのは、書いた人の権利とレイアウトをつけた人の権利、この著者とこの著者をくっつけ合わせてこんなものを作ると非常におもしろい出版物ができるというような企画をした編集者の権利、さらにはさきほどもちょっといいましたが、一部しか読み取らないような場合における読者のフラグメント・アクセスの権利、そういったものが社会的にうまくバックアップされてくると非常に面白い出版形態ができるという期待もっています。

森 今のお話にも関連するのですが、手書きの原稿をみている範囲では、その方が直接書かれているということがよく分かるんですが、最近ワープロ出力の原稿になってきて、そのオリジナリティの評価というのが非常に難しくなっている。それと簡単にいろんなファイルからもってこれる状況になっていますので、最初それを作られた方の著作権を尊重するということがかなり大事になってきている。ソフトウェアを扱っておられる、つまりプログラムを使っておられる方の感覚と、出版に携わっている者の感覚が若干そのへんでずれがあるように感じているわけです。いずれにしても創造活動を促していくという点では、最初に作った人の労力の評価と尊重ということがきわめて大事で、これは文化を高めていく上でも不可欠だということを申し上げておきます。

小町 モディファイしていく人の権利も非常に重要

であると思います。おそらく新しい権利じゃないかとも思います。

森 はい。音楽著作権に比べると、少し出版側のほうが出遅れているんですけれども、法制化の準備がなされています。

司会 余談ですけども、この間あるところで聞いた話ですが、お役所から補助金を貰う仕事をしますと報告書を出すわけです。けれども、報告書の原稿というのは、必ず手書きじゃないとだめなんだそうです。あとで監査があったときに、ワープロじゃだめだといわれるので、ある方がきれいな原稿を作ろうと思って、わざわざ手書きをワープロで打ち直して出されたら、それじゃだめだということで、もう一度手書きに直したという話があります。今のお話で手書きというのは書いた人が本当に書いたというのが分かるという話は、かなり本質的な問題を含んでいるかもしれないですね。

どなたかご質問のある方いらっしゃいますか。

質問 日本ユニシスの若鳥です。質問は複数の方に関連する質問をまずさせていただきます。石田先生に質問したいのは、今 Postscript で取りあえず実現可能なところでいろいろ研究なさっているわけですが、将来は SPDL でも変換系をご用意なさるのではないかなと思うんですが、いかがでしょうか。もう一つ春田さんのほうに、プリントサービスというのが SC18 のダブルビジョンで行われていますが、データ列として SPDL をサポートして、それをプリントサーバで片付けてくれるような可能性はいかがでしょうか。以上です。

石田 私はとりあえずは Postscript を使うということで、将来はたぶんもっと一般的なものということではメーカーさんの製品を期待することになるんじゃないかなと思うんです。というのはこういう問題はちょっと突っ込みますと、非常に複雑になってしまう。たとえば漢字フォントは、本当にちょっと足を突っ込みますと、ものすごく複雑になりますから、大学なんかの小さなチームでやるのは非常に難しいですね。ですから将来はどうなるか分かりませんが、とりあえず Postscript ということです。

春田 印刷サーバのご質問なんですが、さきほどちょっと議論がありましたけれども、印刷サーバにどのような機能をもたせて、それをユーザ側に見せるか、たとえばこういう印刷の言語とかフォントについて、サーバはサポートしていないからそういう要求は受け

られないよという点についてはいまだ決まっています。基本的な方針としては SPDL のような標準化された言語については、極力ジョブ依頼のレベルで印刷可否の判定をして、ユーザに返すことを今考えています。

司会 ほかにご質問おもちの方はおられますか。

若鳥 それでは名言を吐かれた 2 人の方に質問したいんですが、森さんからは、正しい日本語を書ける人はいないとお叱りを受けました。たしかにそうであろうと思います。それから中島さんは文章を書く人が割付けを楽しんでしまうといわれました。DTP をみんな楽しんでいるということなんですが、この二つを合わせますと、割付けというような今までですと、出版の方々の知識に属するようなこと、それから文章でいえば、もともと文章の支援をするというようなこと、そういったことが DTP の知識のデータベースあるいは知識処理によりだんだんにサポートしてもらうということになるのではないかと思うんですが、そのへんの感触についてお 2 人の立場からお願いします。

森 どうも舌足らずでいいすぎたようでして反省しております。実はワープロの原稿がいただけるようになってから文章の練り具合というのは非常に高くなってきているということも事実です。これはおそらく手書きで執筆されると、まあこのへんでといって手を離される歩留りがある。それがワープロで作っていると、何回も読み返して手直しが簡単にできますから、おのずから推敲の度合が高まる結果、そういう意味では非常にいい文章になる。しかし、ちょっと味気ないというんでしょうか、筆が走るということがなくなって、均一化されていくという感じもします。

中島 かという私自身もかなりレイアウトで遊ぶということをしています。たとえば今回お出しした資料を打つときにも、素人ながらどのくらいの行間がみやすいだろうとかということをやりながら、たぶん文章を打った後に 7~8 回プリンタに打ちましたかね。それくらいの無駄をしまして、自分ながらも反省しなければいかんと思っています。

レイアウト情報の管理とかが、もう少し共通化できれば、レイアウトそのものを商品として売る市場は必ずあると思います。そして一般の人はそれを買って使ったほうが自分でよけいなことをするよりきれいだという世界がはるかに望ましいと思っています。

司会 それではほかにご質問をおもちの方いらっしゃいますか。

質問 電総研の岡田と申します。卓上出版とネットワークということで、卓上出版に強い方がいろいろお話されたんですけども、卓上出版という観点からみて、通信系への要請とか、要望とか、そういったものがあればお聞きしたいと思います。

司会 なかなか難しい問題が出ましたけれども、春田さんどうでしょうか。

春田 今、通信ということをやいますと、1対1のだれそれに話をするというような通信はもう日本中、あるいは世界中だれとでも通信ができて、まあ相手がいるかどうかは別として、電話番号を廻せばだいたいつながるようになっています。

ただ、ここでいっているようないろんなアプリケーションを載せた場合の通信ということを考えた場合に、本当に単につなぐだけでいいのか、ここでみえますとそれ以外のいろんな要求条件があると思います。

一番大きな問題は、必要な情報が欲しい人にタイミング良く伝わるネットワークをいかにして作るかということだと思います。それについてはどうすればよいか一概にはいえないんですけども、われわれとしては、ネットワーク側の機能をどんどんふやして、なるべく通信会社として端末側とお互いに仲のよい機能分担をいたしまして、全体としてより知的といいますか高機能な通信ネットワークを作っていきたいと思っています。

司会 中島さん、何かございませんか。

中島 私のほうでいうと、ちょうど LAN のほうからみて発展型という形ですから、逆の立場だと思います。たとえば SQL のデータベースサーバのことを考えますと、たぶん実行形態としてはローカルにネットワークでつながっているワークステーションからトランザクションを起こして照会にいくと、それでデータが得られれば満足しますけれども、そのネットワークの中になかったときにどうするかという話が、議論として出てくると思うんです。その場合 SQL がちゃんと広がってくれば、LAN から外に出ていって構わないわけですね。そのときどこへたどりつけばいいかというインフォメーションをどう管理するかということは、どちらの分担かというよりも、双方で解決していかなければならない問題だと思います。そういう意味でのディレクトリ・サーバみたいなものが今後必須になってくるだろうということを感じています。それが交換機レベルにあるのがいいのか、ワークステーションにあったほうがいいのか、その場合のワークステー

ションのディレクトリ情報がどうやって、アップデートされるかなど、研究していく余地はあると思います。

司会 時間もそろそろ迫ってまいりましたので最後に、ひとことずつどうぞ。

石田 メーカのみなさんへの希望を一つ、最近いろんなメーカさんから卓上出版の専用システムというのが発売されています。しかし狭い意味での卓上出版の技術者はネットワークのことをまったく知らない、それからネットワーク屋さんはもちろんそういうところに入ってこないということがあるものですから、社内ですべてのへん緊密な協力体制をしいていただきたい。やっぱり卓上出版システムというからにはネットワークにつなげることが絶対必要なんです。ネットワークの中であちこちで作られた文書情報を集めてきてだれかが編集する、そういうのが一番いいだろうと思います。

それについては中島さんも私も非常に気になっているのは、レイアウトに凝りだすと、本当にそのために時間をくってしまって、有用な仕事ができないという可能性があることです。文章を作る方はレイアウトは全然気にしないで、生のテキストだけ作って、ネットワークを介してそれを卓上出版専用のシステムに送ると、それにレイアウトを専用にする人がいて、ちゃんとした文書を作ってくれる。やっぱり各職場にそういう体制を作らないと、みんながでんでんばらばらにレイアウトしていたのでは、非常に時間の浪費になるんじゃないかと思います。そういう点からすると、研究者にとって比較的いいんじゃないかと思っているのは、Tex というシステムです。あるいは Troff でもいいんですが、まあ Tex のほうが機能があると思うんですが、Tex にさらにオブジェクトを被せたもので LaTeX というのがあるんですけども、これが最近日本語化されております。これですとこの部分はタイトルであると言いつて、タイトルを書く、それからここは著者名ですよと言いつて著者名を書く。それからあと、ここは第1パラグラフのスタートということをやっちゃうと言いつて文章を書きはじめる。それだけで最終的に比較的きれいな文書が出てきて、レイアウトの細かいことはあまりいわなくていいということですから、研究者が自分でやるんなら日本語 Tex みたいなものが、おそらく一番いい。これは画面でみたおりのものを出すとはなっていないんですけども、手間を省くことからすると、たぶんそれが一番いいのではないかと思います。

次に、レイアウトなんですけれども、パソコンとかワープロのユーザ全員が自分でレイアウトするというのは、たぶん非常に無駄なことだと思いますから、各職場でそういう担当を決めて、その人をうまく教育することが必要じゃないかと思いますね。

あと研究テーマとしては、これから大勢の人が集まって力をあわせて文書を作る。これは報告書を作るとか本を作るとかというときに、よく出てくると思うんですけども、それを支援するためのシステムがぜひ必要ですね。大勢でやるとなると必ずネットワークが必要です。日本でちょっと心配なのは、そういう研究を真面目にやっている人があまりいないような感じがすることです。その点はアメリカはさすがでグループでもって大きな文書を作るときにどうしたらいいかという研究がずいぶん進んでいるようです。そのへんは日本でもやらなければいけないんじゃないかと思っています。

司会 それでは小町さん何かございますか。

小町 今、石田先生からレイアウトする人がいないというような話を承りましたが、これはやはりレイアウトする処理系、編集をする処理系、それらに人間がアクセスするためのユーザシステム・インタフェースが非常に難しいというところに、一つの基本的な問題点があるような気がします。そのようなユーザシステム・インタフェースを ISO では WG 9 というところでやっているんですが、まだ使いやすい文書編集系、あるいは文書レイアウト系のユーザシステム・インタフェースを提供するところに至っていません。ISO は今後かなり積極的にその標準化作業を進めていく必要があると思います。

それから最後に質問された通信系に関してのコメントを一言。パッケージメディア、特に CD-ROM での出版というのは、すでに始まっておりまして、それなりの評価を受けています。この CD-ROM 出版の一つの問題は、ユーザレベルでの書き込みができないということです。私、今ドキュメントをみながらみなさんの話をメモして、この上に書き込んでいるんですが、そういった機能はぜひ欲しいと思います。パッケージメディアにパーティションを切って、ある部分から先は書き換え可能メモリである、少なくとも1回は書き込めるというような機能が、今後パッケージメディアに対して要求されるのではないかと思います。

それから、今ヨーロッパ、アメリカあたりで SGML の普及が始まろうとしておるんですが、日本ではいま

いちその動きがないという問題があります。アメリカではヨーロッパもそうなのですが、マークアップの定義が明確にされていて、たとえば MIL, AAP のマークアップが使える状態にあります。日本ではまだこのへんが未発達でして、なんらかの動きを期待しています。

春田 電子出版の定義について、森さんが定義されていて、第一から第三までございまして、私は第一出版ぐらいの話を出版会社の方は考えておられるのかと思っていましたら、第三電子出版というところまで考えておられる。これはよくよく考えてみますと、ネットワークを使いました一種の付加価値ネットワークをめざされているような気がしますので、これからは通信会社と出版会社とをもって仲良くいたしまして、より良いサービスを作っていく必要があるという感じがいたしました。

それでこういう第三出版が世の中に認められるようになりますと、おそらく今の本屋さんの小売店とか問屋さんにデータベースをおきまして、それをみんながパソコン通信でみにいくようなことになるかと思えます。まあ本屋じゃなくても電話局においてもいいんじゃないかと、そういうこともちょっと考えたりしておりますので、今後面白いビジネスが期待できそうだと思います。

森 私どもも、第三電子出版ということで、一応パソコン通信で OHM-NET を実験稼動しております。これは科学と技術を語り合おうということをテーマに、ネットワークを通じた出版のあり方を追求しています。これについてはまだ報告する段階に至っておりません。レイアウトに関して一言。今日いらっしゃる方々は、出版サイドというよりは、むしろ一般コンピュータ・オフィス関係の方々じゃないかと思うんですが、午前中に日本語処理の高品位文書処理のお話がございまして、やはりフォントを充実させていくとか、見映えを良くしていく、それからフォントの種類

を多くしていくなど、相当カバーしなければならないものが多いわけです。それらを均一にカバーしようとすると、相当量のメモリを要するし、速度そのほかにも影響してくる。実際のレイアウトというのは単純なほど良いと考えています。さきほど石田先生の最初の画面で Postscript で作成した HAL の非常に鮮やかに凝った図を紹介されました。普通の方が画面に向かって、時間をとられるというのは、そのへんをあれこれ工夫をしているんなものを使いたいという欲求にかられて、時間を費すんじゃないかと思えます。あのような HAL というのは一つの文章に一個あればいいんですね。あれを何度も使うと、今度はあきがくる。今後レイアウトをされる方は一番単純なレイアウトでワンポイントだけ効かすというあたりがよろしいんじゃないでしょうか。

中島 今回パネラということで参加させていただきましたけれども、大変いろいろ面白い話が聞けて勉強になったと思います。印象に残ったのは、まず電子出版という言葉についてで、私も、第一と第二というのはかなり異質のものなのに、世の中では同一に使われているということに不満をもっていましたので、ひょっとしてこう場で、第一電子出版、第二電子出版という言葉が標準になってしまえばいいなと思いました。

もう一点は、レイアウトについての議論ですね。たとえば私の原稿ですけれども、英文のタイトルは一番下にというふうにいわれたんですが、ワープロの都合上できなくて、間に入ってしまった。たとえばこういうことも、この原稿の話でいえば私どもから SGML のドキュメントがいくと、情報処理学会のほうで、それをレイアウトしていただけるというふうな時代が2年か3年後ぐらいにあればいいなというふうに思いました。

司会 それでは時間も少し超過したようでありますので、これで本日のパネルディスカッションを終りにしたいと思います。