

エンドユーザーは 帳票システムに何を求めているか カスタマイズ機能を視野に入れる

大澤 文孝
OSAWA, Fumitaka

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:

Level



Samples

はじめに

近年のエンドユーザーは、ワープロや表計算ソフトに使い慣れているため、業務アプリケーションの帳票機能にも、それと同等の機能を望むことが少なくありません。

その一方で、開発者にとっての帳票は、アプリケーションの中核ではなく、さして重要な部分ではないと考える節があります。

実際、業務アプリケーションはグラフィックソフトではありませんから、重要なのは、ロジック通りに動作するかどうかであり、美しく最終的な印刷物を作るのは主目的ではありません。

そのため開発者は、帳票機能の実装に、あまり時間や手間をかけたくないというのが本音でしょう。

もちろん、十分な時間がありさえすれば、エンドユーザーの要求通りの帳票を作ることができます。しかしそれは、開発期間、そして、開発コストが増すということを意味し、それがエンドユーザー（そして発注者）のために

なるとは限りません。

そのため帳票の実装は、「いかに手間をかけずにエンドユーザーの要求に応じていくのか」がポイントになると言えるでしょう。

そこで本稿では、エンドユーザーが帳票にどのような機能を望んでおり、どのようなアプローチで解決していけばよいのかを考えていきます。

帳票の難しさ

帳票とは、印刷面の指定された場所に特定のデータを埋め込んで印刷する、いわゆる「差し込み印刷機能」です。

原理的に言えば、集計したデータなどを特定の位置に印刷するロジックをアプリケーションに組み込めばよいこととなります。

もっとも原始的な実装としては、APIを使ってプリンタの印刷面を得て、指定された位置にデータを文字として描画すればよいということになります。

具体的に.NET Frameworkで言えば、PrintDocumentオブジェクトを作り、

その描画ルーチン内で、DrawStringメソッドを使って、文字列を描画することになるでしょう。

このように述べると帳票は簡単に思えますが、実際には、用紙サイズに収まるようにするため、いくつかの例外的な処理が必要になります。

可変文字長の問題

差し込まれるデータの長さが、いつも一定長以下ならば、あらかじめ印刷面に埋め込まれる位置や大きさを固定できます。

しかし実際には、データの長さがまちまちで、データの内容によって、定めた大きさに入りきらないことがあります。

そこで、入りきらないときには、次のような実装が必要になります(図1)。

①無視してトリミング

入りきらない部分を無視して、印刷しないようにします。実装は簡単ですが、エンドユーザーは、これを許さないでしょう。

②無視してそのまま描画

はみ出た部分もそのまま描画します。これも実装は簡単ですが、他の場所に文字がかかったり、最悪、用紙サイズを超えたりすることになるので、やはりエンドユーザーは、許してくれないでしょう。

③縮小して押し詰める

文字の大きさを小さくして強引に押し詰めます。実装にあたっては、どの程度文字を小さくすれば収まるのかを計算するロジックが必要になるので少し複雑になります。とくにあまりに文字数が多すぎると、目に見えないほど文字が小さくなってしまふ恐れがありますから、その配慮が必要です。縮小する場合には、文字の大きさを小さくする方法だけでなく、横幅だけを縮めて長体をかける方法もとれます。

④折り返す

入らない場合には、折り返して2行以上にして描画します。

行が増えるので、今度は、垂直方向に、はみ出してしまう恐れもあります。そこで、③の方法と組み合わせて、折り返しつつ文字を縮小して納める実装も考えられます。また日本語では、句読点などが行頭に来ることは許されないという「禁則処理」の問題もあり、どこで折り返すのかを決めるのが難しい側面もあります。

多くの場合、望ましい実装は、③か④ですが、どのような用紙に印刷するのか、また、どの程度はみ出るのかによって異なってきます。

はみ出してしまうのは、物理的に仕方ないので、「いかに破綻していないように見せるのか」が帳票実装の腕の見せ所になります。

入力できる文字長は帳票サイズから決定するのが望ましい

このように、文字がはみ出すことまで考慮すると、帳票の実装が複雑になってきます。

そこで、すでにデータの時点で、「このデータは何文字以上には、なり得ない」と決めうちできるならば、そうすべきです。

たとえば、「住所」は、「住所」という1項目のデータとして入力するのではなく、「都道府県」「市区町村」「番地」「ビル名」などに分け、それぞれ決まった文字しか入らないようにしておいたほうが、帳票の実装が簡単になります。

同様に、「摘要」などのメモ的要素をもつ欄も、あまりに長い文字長の入力を許すと、帳票印刷のときに、はみ出す心配が出てきます。

発注者は、このあたりのことを知るよしもないので、

開発者：「この欄は、どのぐらいの長さが必要ですか？」

発注者：「今は、XX文字も入れれば十分ですが、余裕をもって長いほうが良いですね」

といったやりとりで、あまりに余裕を持たせると、帳票印刷で、はみ出し処理を考慮しなければならなくなる可能性があります。

もちろん、今後の拡張を考えれば、

図1：幅に入りきらない場合の処理

①無視してトリミング

東京都新宿区舟町5 株式会社翔

②無視してそのまま描画

東京都新宿区舟町5 株式会社翔泳社 出版局

③縮小して押し詰める

(a) フォントの大きさを小さくする

東京都新宿区舟町5 株式会社翔泳社 出版局

(b) 長体をかける(幅だけ縮める)

東京都新宿区舟町5 株式会社翔泳社 出版局

④折り返す

東京都新宿区舟町5 株式会社翔泳社 出版局