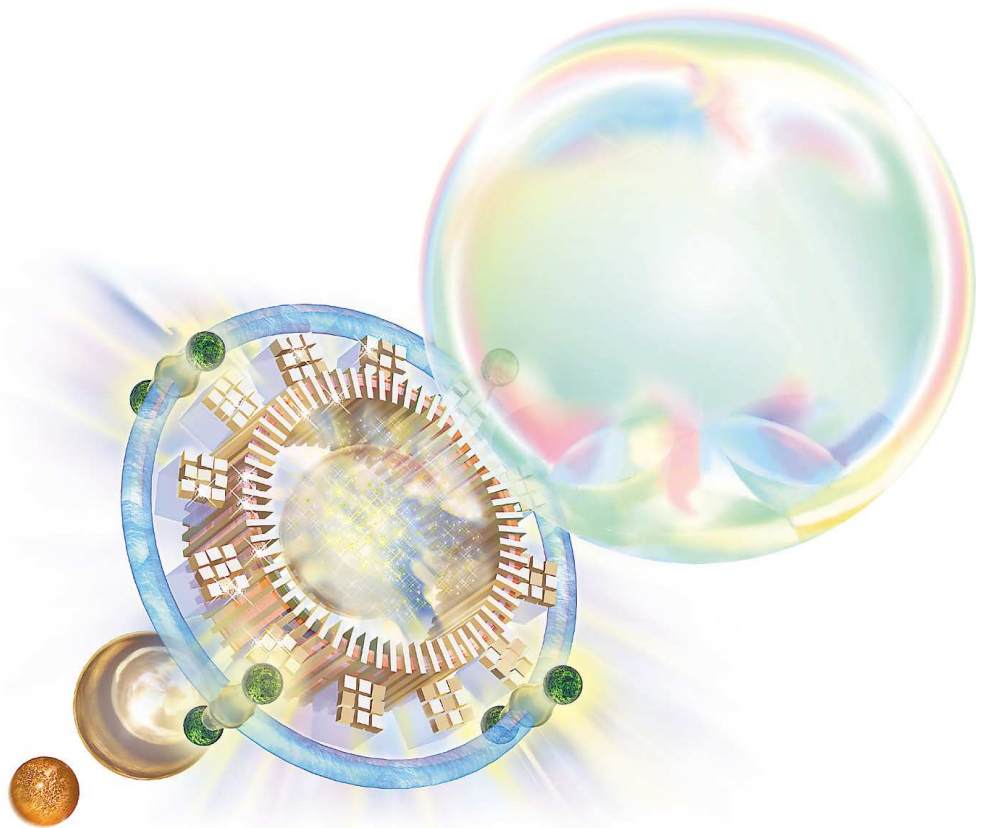


SVGコンバータ

for DXF
for HPGL
for Visio

株式会社マイスター



XML対応製品をレビュー

Product & Solution

プロダクト&ソリューション



既存の図版リソースをWeb上で再利用可能とする
SVG変換アプリケーション

SVGコンバータ シリーズ

株式会社マイスター

■開発経緯

株式会社マイスター(以下マイスター)では、以前からAutodesk社のAutoCADやMicrosoft社のVisioを利用した建物の施設管理(Facility Management)システムを開発している。昨今の一般企業へのインターネット・イントラネットの普及により、顧客から「ネット上でCADソフトウェアなどを利用して作成した図面を閲覧できないか?」また「高価なソフトウェアを購入しなくても閲覧可能にならないか?」など多くの要望が寄せられている。そのため、より最適な手段を模索していた。

マイスターがSVGに注目し始めたのは約1年前のことで、当時SVGはまだワーキングドラフトの段階であったが、SVGというものが実際の程度使えるものなのか検証を行なうために、SVGの実証試験に着手した。その結果、実用性と将来性について十分な結果が得られたためSVGコンバータの開発が開始された。

従来、CADソフトウェアで作成した図面を閲覧するためには、図面の作成者が使用したCADソフトウェアやその図面専用の閲覧ソフトウェアが必要であった。そのため、ただ単に図面を見るだけの理由では高価なCADソフトウェアを購入することもできないし、フォーマッ

トがソフトウェアによって異なっているので統一されたビューワーで見ることができなかったのである。しかし、ファイル仕様をSVG化することによって、安価(Adobe社が「SVG Viewer」を無料で配布)でかつ、統一されたビューワーで図面を閲覧することができる。SVGは図面をネット上に持ち込む最もスマートな手法であり、SVGコンバータはその普及を促すソフトウェアである。

■商品概要

SVGコンバータ for DXF

「SVGコンバータ for DXF」(以下DXF版)は、「SVGコンバータ」として、本年5月にリリースされたシリーズ最初の製品である(図1)。

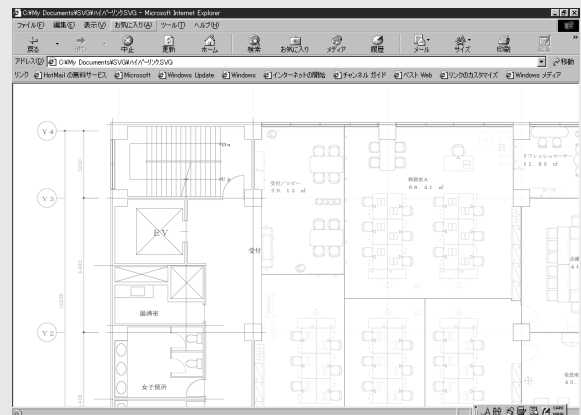
DXFとは、Autodesk社が開発したCAD図面交換ファイルの標準的なファイル仕様で、現在多くのCADソフトがDXFファイルの書き出し機能を持っている。

SVGの仕様では図形が2次元までの扱いられているので、DXF上の3次元の図形データ(Z座標)は無視す

る形で変換されるが、一般的に平面図や断面図といった図面の多くが2次元図面として作成されているため、現在ある多くのCADデータをネット上で公開することが可能であり、既に多くのユーザーに利用されている。

また現在DXFについては、Autodesk社以外のCADソフトウェアで解釈などの違いがあり必ずしも仕様通りのファイルにならない場合がある。その結果他のCADソフトウェアで読み込めない、図形が崩れるなどの問題が発生しているが、DXF版ではDXFファイルを読み込む時点でファイルを正規化した後、SVG化するので読み込みエラーや変換ミスなどの発生が軽減されている。

図1: 「SVGコンバータ for DXF」



SVGコンバータ for Visio

「SVGコンバータ for Visio」(以下Visio版)は、本年12月にリリース予定のソフトウェアで、Microsoft社の「Visio 2002」用のプラグインだ(図2)。Visio 2002はUML、フローチャートやネットワーク図といったエンジニアリング系の図面はもとより、オフィスレイアウト図やスケジュール管理図など、現在幅広い業種で利用されているドローイングツールである。また文字調整やグラデーション処理といったデザイン機能を有している。Visio版では図形以外に文字調整の変換、Visio上で設定したハイパーリンクの変換など、VisioをSVGエディタとして利用できるように開発を進めている。

SVGコンバータ for HPGL

「SVGコンバータ for HPGL」(以下HPGL版)は、シリーズ3本目の製品で、本年12月にリリース予定である(図3)。Hewlett-Packard社が開発したプロッタやレーザープリンタで利用されているグラフィック記述言語HPGLをSVGに変換する。多くのCADソフトウェアはHPGLファイルの書き出し機能を持っており、HPGL版では特に3次元図面でのアイソメ図(2.5次元)をHPGL形式へと変換することによってネット上でアイソメ図の図面を扱うことが可能となる。

現在HPGLにはHPGL/2と呼ばれる拡張規格があるが、HPGL版では今のところ対応していない。

コンポーネントソリューション

DXF版およびHPGL版は単体商品のほかに、ベンダー向けとしてCOMコンポーネント版も用意している。そのため他のシステムへの組み込みなども可能と

なっている。

■SVGの実際

SVGコンバータでの取り組み

SVGを実際に利用してみると、実にさまざまなグラフィック表現を的確にまとめてあることが分かる。以下では、SVGコンバータが実際にSVGを生成する際に活用した一例を紹介する。

座標系の違い

VisioやDXFなどで各頂点の座標が(20,10)、(40,50)、(60,20)である図形を表現すると図4のようになる。これを単純にSVGで記述し直して表示させると図5のように上下が反転された状態になる。

一般的なCAD図面などの座標系は図4

図2: SVGコンバータ for Visio



図4: VisioやDXFなどで作図した図形

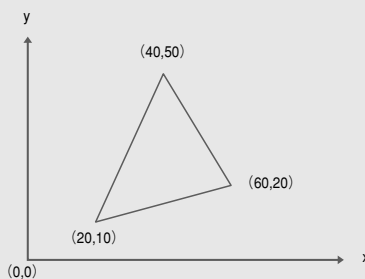


表1: 座標変換の指定

属性	解説
rotate	回転
scale	拡大、縮小および描画方向の指定
transform	移動
skewX/skewY	歪み

に示すようにy座標が上方方向に増加するようになっている。ところが、SVGの座標系は図5のようにy座標が下方方向に増加するようになっている。

これは、SVGがWebでの活用を念頭に設計されているからだと考えられる。結局のところWebの画像を表示するのは一般的にPCであり、PCの座標系のほとんどが図5のようになっているからである。

これを解決するためには、座標系の変換が必要になる。SVGは座標系を変換するためにtransform属性を持っている。transform属性には、表1に示すようなものがある。

y座標の向きを変えるには、scale変換を利用する。scale変換を適用したサン

図3: SVGコンバータ for HPGL

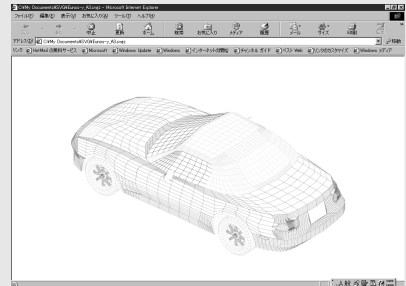
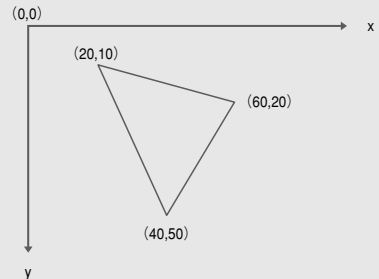


図5: SVGで作図した図形



■リスト1: scale変換を利用するサンプル

```
<g transform="scale(1, -1)">
  <path d="..." />
</g>
```

ブルはリスト1のようになる。

scaleの尺度を1とし、y方向にマイナス(-)を指定することでy座標の向きを反転させることができる。また、各図形(この例では<path>エレメント)にも直接scale変換を指定することはできるが、リスト1のように各図形のコンテナである<g>エレメントに指定することで、その内部にある図形は座標系の違いを意識することなく、そのままの座標で記述することができる。

ブロック図形への対応

DXFには図面上の各種図形情報を格

納するセクションとしてENTITIESセクションがある。

通常、円や線などの図形は、このENTITIESセクションに格納されている(図6)。しかし、図面に描かれている図形のうち、同じ形ものがいくつも存在する場合に、一度描いた図形を座標の違う場所にいくつも配置することができるように「ブロック図形」というものが定義できるようになっている。

ブロック図形自体はBLOCKSセクションに格納されており、その中に実際の図形(CIRCLEやLINEなど)が格納されている。

さて、そのブロック図形を利用する目的でENTITIESセクションにあるのがINSERTオブジェクトである。このINSERTオブジェクトは、ブロック図形に対するリンク情報を持っている。INSERTオブジェクト自体は図形ではなく、ブロック図形をどこへどのように配置すればよいのかという情報(座標など)のみを持っている。

SVGにも、このブロック図形のような仕組みが用意されており、SVGコンバータではこれを採用している。SVGでは、DXFのBLOCKSセクションに当たる部分を<defs>エレメント内に記述する(図6)。<defs>エレメント内に記述した図形は、そのままでは表示されないようになっている。

この例では、BLOCKSセクションのBLK_Aオブジェクトに相当するものとして、<defs>エレメント内にID="BLK_A"と指定した<g>エレメントを作成し、その内部に各図形を記述する。

これを、表示対象の<g>エレメント内で<use>エレメントによって参照し、指定座標に表示できる。

DXFファイルの図面内容(ブロック図形の使用頻度)にもよるが、この方法を使うことでSVGファイルのサイズが大きくなるのを抑えることができる。

図6: DXFとSVGのブロック図形への対応

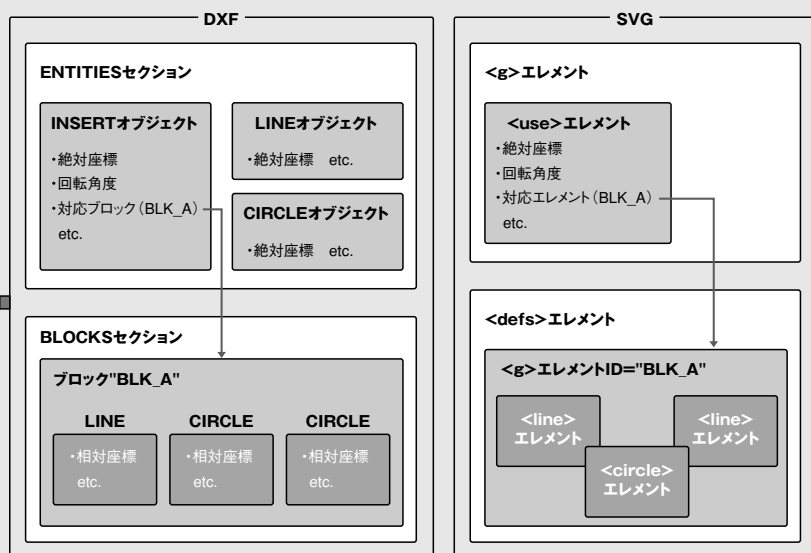


表2: HPGLコマンドの解説

コマンド	機能	リスト2の解説
IN	プロッタの初期化	プロッタの各種設定を既定値にする
SP5	ペンの選択	ペン番号5にセットしてあるペンの色で描かれる
PA20,10	ペンの移動	指定座標位置へペンを動かす (この時点ではペンが下がっていないため描かれない)
PD	ペンを下げる	描く準備
PA40,50	ペンの移動	直前の座標から現在の座標まで線(ひとつめの辺)を描く
PA60,20	ペンの移動	ふたつめの辺を描く
PA20,10	ペンの移動	3つめの辺を描く

リスト2: HPGLコマンド

IN; SP5; PA20,10; PD; PA40,50; PA60,20; PA20,10;

HPGLで図形を描く

HPGLを使ってプロッタに図形を描かせるためには、通常HPGLコマンドが記述されたプロットファイルのプロッタに渡す。それを受け取ったプロッタはHPGLコマンドをひとつずつ実行して行きながらグラフィックを描く。

例えば図4に示した図形を描画させるためのHPGLコマンドを表わすと、リスト2のようになる(詳細なコマンドは省

略)。コマンドの解説は表2に示す。

ちょっと見ただけでは、何がなんだか分からないコマンドが並んでいて、かなりハード（プロッタ）を意識した記述になっていることが分かるであろう。しかし、プロッタが理解できれば問題はない。

SVGで描く

さて、リスト2のHPGLコマンドを、SVGに置き換えるとリスト3のようになる（詳細は省略）。

<path>エレメントは、直線や曲線を表現するためのもので、プロッタのような線の描画には適している。ちなみに、<path>エレメントの d= の後に記述されている部分がパスのデータで、コマンドを含んでいる。コマンドを簡単に解説したものを表3に示す。

テキスト検索への対応

テキストをSVGで表現する場合、<text>エレメントを記述する。

最も単純なテキスト表示のためのSVG記述サンプルはリスト4のようになるが、例えば、図7のようなテキストを表現するためにはリスト5のように記述する。

このように記述されたSVGの場合、文字列の検索に制限が出てくる。この場合、"Sample"という文字列を検索しても、ヒットしなくなる。SVGでは<text>エレメントで囲まれた部分を検索文字列の単位として扱う。そのため、サンプルのような<text>エレメントを記述した場合、"Sam"と"ple"というそれぞれ個別の文字列として扱われる。

これを"Sample"という文字列で検索できるようにするためには、リスト6のように記述する（リスト6のほかにも、さまざまな方法がある）。

<tspan>エレメントは、<text>エレ

リスト3：HPGLコマンドをSVGに置き換えた場合

```
<path d="M20,10 L40,50 L60,20 Z" style="fill:none;stroke:#0000ff;stroke-width:0.1" />
```

表3：pathコマンドの解説

コマンド	機能	リスト3の解説
M20,10	移動コマンド	絶対座標(20,10)にカレントポイントを移動する
L40,50	線の描画	絶対座標(40,50)まで直線を描画する (カレントポイントが(40,50)になる)
L60,20	線の描画	絶対座標(60,20)まで直線を描画する (カレントポイントが(60,20)になる)
Z	パスのクローズ	パスを閉じる (カレントポイントからパスの最初のポイント(20,10)まで直線を描画)

図7：SVGサンプル



リスト4：<テキスト>エレメント記述例

```
<text x="30" y="40">Sample</text>
```

メント内の文字列の特定の部分に対して属性の変更を指定する場合に利用する。

このようにすれば、図7のように見かけ上は分断された文字列でもひとつの文字列として扱うことができるようになる。

また、<tspan>エレメントごとにx,y座標を指定することができるので、文字間の調整なども行なうことができる。ただし、<tspan>を使わなくても単純な文字間の調整を行なうためにletter-spacingなどの属性も用意されているので、ケースに応じて使い分けができる。

■SVGの可能性

WebとSVG

SVGの応用例としてよく語られるの

リスト5：単純な記述例

```
<text x="30" y="40">Sam</text>
<text x="80" y="30">ple</text>
```

リスト6：図7のように記述する例

```
<text>
<tspan x="30" y="40">Sam</tspan>
<tspan x="80" y="30">ple</tspan>
</text>
```

は、Webシステムにおけるダイナミックなグラフィック生成である。SVGコンバータが提供しているのはスタンドアロン環境におけるSVGの生成であるが、このような機能をサーバー側が備えるだけで、ビルや施設の情報を管理しているシステムなどでは（グラフィックの表示方法によってはデータベースの内容を見直す必要があるかもしれないが）、よほど複雑な表示でもしなない限り実現できることが理解してもらえたであろう。

SVGコンバータ シリーズを開発したことで、SVGはその真価を問う段階から、Webシステムなどへの応用に向けて、ますます具体化が進んできたことを実感している。