

# savic-netFX Webビル監視装置の開発

## Development of savic-netFX web building automation system

株式会社 山武  
ビルシステムカンパニー

山本 博之  
Hiroyuki Yamamoto

### キーワード

savic-netFX、システムマネージメントサーバ、SVG、Java、XML、PDF

近年のWeb技術の発達にともない、ビルディングオートメーションシステムの監視装置は中央監視室で集中して監視する形態に加え、Web技術を利用し様々な場所で監視する運用も求められるようになってきた。一方Webアプリケーションで従来の監視装置と同等の機能を実現するためには、機能にあったWeb技術を選択し活用していくことが必要であった。本稿ではsavic-netFXで実現したリアルタイム監視機能、ユーザーインターフェース、印刷機能の仕組みを機能ごとに紹介する。

Previously, the monitoring of building automation systems was done with the monitoring devices concentrated in a central monitoring room. But with the development of web technology in recent years, there are now also demands for operations in which monitoring can be done from various locations through the use of web technology. Meanwhile, in order to use web applications to achieve the same functions as the conventional monitoring devices, it had been necessary to select and use the web technology that suited the functions. In this paper, we introduce the mechanisms for each of the real-time monitoring, user interface and printing functions, which have been achieved using savic-netFX.

### 1. はじめに

専用の中央監視装置を必要としていたビルディングオートメーションシステム（以下BAシステム）は、近年のWeb技術の発達にともない形態を変化させつつある。savic-netFX（以下FX）はこのような流れの中Web技術の利点を従来の中央監視装置に取り入れ開発したシステムである。

FXではWebサーバを持ち、ユーザーにBAシステムの監視画面を提供するデバイスとしてシステムマネージメントサーバ（以下SMS）を開発した。

SMSは監視点数30000オブジェクト、100台までのデバイスが監視可能であり、同時に5台のクライアントPCに監視画面を提供する。ネットワークはIPv4だけでなくIPv6のネットワークをサポートし、プロトコルとしてBACnetを標準サポートすることにより他社との接続性を向上させている。ビル監視

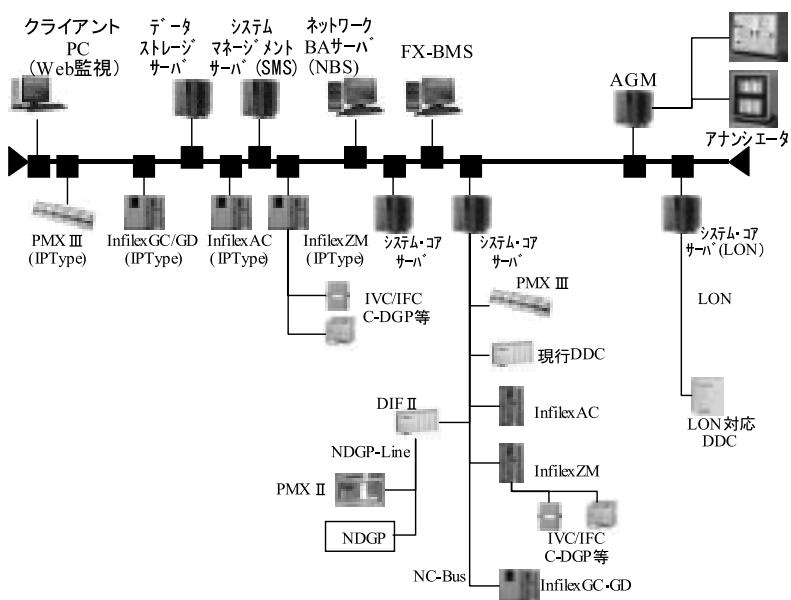


図1. savic-netFX システム構成

装置のネットワークと通信可能なネットワーク上にクライアントPC上を接続することにより、ブラウザ上で監視画面を表示することが可能となっている。

SMSの提供する画面はブラウザ上のWeb監視画面ではあるが、24時間監視の信頼性、リアルタイム性、表示速度性能は従来の監視装置と同等の性能で設計されている。またFXの名前の由来でもある(Flexible)柔軟で拡張性豊かな設計となっている。このような特徴を持つFXシステムのWeb監視に関する技術を紹介する。

## 2. リアルタイム監視

BAシステムの重要な機能として、警報や状態の変化などビルの時々刻々と変化する状態をリアルタイムにユーザーに通知する機能が挙げられる。

FXのリアルタイム監視機能としては図2に示すような新規警報表示機能がある。新規警報表示機能は警報発生後、即座にユーザーへ警報情報の通知およびブザー鳴動を行う。



図2. 新規警報表示

FXでは警報通知機能以外にもポイントの状態表示画面などにおいてもリアルタイムに情報を更新している。以下ではWebアプリケーションでどのようにリアルタイム監視機能を実現しているかを説明する。

ブラウザの仕組みは基本的にHTTP(Hyper Text Transfer Protocol)を利用してサーバと通信する。HTTPは図3に示すように、リクエストを受けてレスポンスを返すプロトコルである。1回のリクエストに応答して、レスポンスのメッセージを1回で返すためサーバ側からデータの更新をうながすことは難しくなっている。

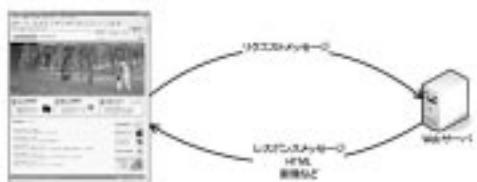


図3. HTTP通信の流れ

この問題を解決する方法としてサーバプッシュ方式と定期デマンド方式の2種類の方法がある。

サーバプッシュ方式の通信の流れを図4に示す。

サーバプッシュ方式はクライアントからの1度の要求に対してHTTP通信の接続は継続したまま、レスポンスマッセージを少しずつ継続的に送信する方法である。

この方式は状態変化発生時にSMS主導でクライアントに情報を送信することが可能である。従ってサーバは状態変化を受信した場合に即座にクライアント側に通知することが可能でありリアルタイム性が良いという利点がある。

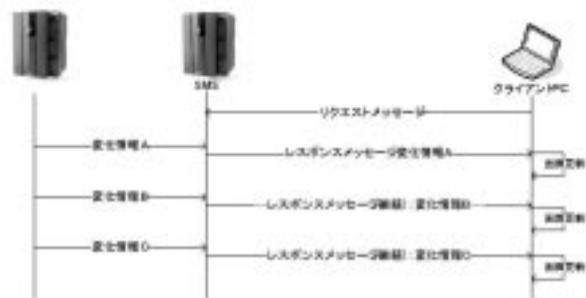


図4. サーバプッシュ方式

一方でクライアントPCとサーバ間のネットワーク障害などが発生し変化通知が伝えられない場合に、クライアント側で異常認識することができない問題がある。これは変化通知がない場合には、SMSとクライアントPC間で通信が行われないため、クライアントPCではSMSからメッセージが通知されていなくても正常と認識してしまうためである。このため、クライアントがサーバと接続が確立できているかの確認を別の仕組みでおこなう必要があり、処理が複雑になるなどの問題がある。



図5. 定周期デマンド方式

次に定周期デマンド方式の通信の流れを図5に示す。

定周期デマンド方式はクライアントから短い周期でアクセスを行ない、サーバ側に蓄積した情報を取得し更新することにより、リアルタイムで更新しているかのように見せる方式である。

この方式はクライアント主導で接続を行うためSMSに異常が発生した場合やクライアントPCとSMS間のネットワークに障害が発生した場合に即座にユーザーに通知することが可能である。しかし、サーバへのデマンド周期によっては警報発生からユーザーへの警報通知が遅れてしまうという問題点がある。

また、変化通知が発生していない場合でも定周期でサーバへのリクエストを行うため、クライアントPCの台数が増加するにつれサーバ側に定常に大きな負荷が発生する可能性がある。

FXではネットワーク異常も含めシステムに異常が発生した場合に、確実にユーザーに伝えることを最重要であると考え、定周期デマンド方式を採用した。

ただし、デマンド周期を極力短くすること、デマンド受信時の処理を極力簡単にすることによりユーザーへの通知がリアルタイム方式と同等となるように工夫している。

この仕組みを図6に示す。

基本的なメッセージの手続きは通常の定周期デマンド方式と同様である。変更した部分は初回のリクエストでサーバ側の最新情報IDをクライアント側に返却(図6-②)している点である。クライアントは2度目の接続以降は、この最新情報IDを付属してサーバに接続(図6-③, ⑦, ⑩)することにより、サーバ側は変化がない場合には処理を行うことなくクライアントに情報なしのメ



図 6. FXの定期周期デマンド方式

メッセージを返却する(図6-④,⑪)。この結果状態変化が発生していない場合のリクエスト受付からレスポンスまでの処理時間は10msec以下で可能となり、サーバ側の負荷も複数のクライアントを接続した場合でもほとんど影響がみられなくなっている。

また変化がある場合でもサーバ側の負荷を軽くするための措置として、初回要求時にクライアントで取得要求情報を通知(図6-①)するようにした。取得要求情報には、クライアントが必要とするポイントの一覧などの情報を格納し、リクエストを受ける前にクライアントへ送信する情報を作成することが可能となった。この結果、1秒に1回のクライアントからサーバへリクエストが可能となり、従来の専用監視盤と同等のリアルタイム性能を実現する事ができた。

### 3.FXユーザーインターフェース

従来、Webブラウザを利用したHTMLクライアントは表現力に制限があり、「エンドユーザーの混乱を招く画面遷移が多い」、「キーボード操作に不向き」など、ユーザーの生産性を下げるといった弱点が指摘されていた。また、サーバに接続するユーザー数が増えるとサーバの負荷増大に伴い画面表示、操作のレスポンスが遅くなり、リアルタイムの監視システムを作る上では大きなデメリットとなってしまう。

この問題を解くため、FXではリッチクライアントシステムのコンセプトを採用することとした。リッチクライアントシステムとはHTMLクライアントの問題点を解決し、機能やユーザーインターフェースの表現力や操作性、表示される内容などが豊かであり、サーバ負荷の削減を可能としたクライアントマシン環境である。クライアントとサーバ間で負荷の分散を図り、サーバ側ではリアルタイム処理を中心に行い、クライアント側で画面作成/描画を行っている。これにより、従来の専用監視盤と同等の表現力、操作感を持ったWeb画面を実現することができた。

このリッチクライアントシステムを構成する技術としては一般的にFlashやJavaテクノロジ、SMILやSVGなどのXMLテクノロジなどである。FXではSVGやJavaScript、JavaAppletを利用して画面表示をおこなっている。

#### 3.1.Java Appletを利用した画面表示

FXのJavaAppletを利用した画面の表示例としてグループリスト画面から、ポイント操作ダイアログを表示した画面を図7に示す。

通常HTMLで作成された画面では表現が制限されたため、タブ表示や、状態に応じたボタンの有効／無効の切り替え、ダイ



図 7. ポイント操作ダイアログ

アログの上下関係を制御するなどユーザーの状況に合わせたインターフェースの提供は困難である。

FXではJavaAppletを利用することにより専用の監視装置と同様のユーザーにわかりやすい画面の構築が可能になった。

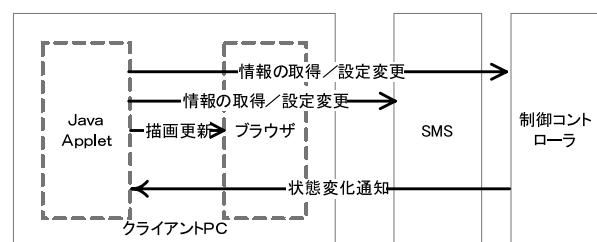


図 8. ポイント操作ダイアログの処理の流れ

このJavaAppletでの画面表示の仕組みを図8に示す。JavaApplet側で画面表示処理を行うことにより、SMSでは情報の提供、制御コントローラへの通信の中継作業のみの作業となりリッチクライアントの利点であるサーバでの作業負荷が少なくなっていることがわかる。

#### 3.2.SVGを利用したサマリグラフ表示

SVGとはXMLベースの2Dベクター画像記述言語であり2001年9月にW3C勧告として公開されている。

ベクター画像は画像を点の集合体ではなく、線や面などの図形の集合体として扱うため、bitmapやjpeg形式のファイルなどと異なり、拡大しても画像が乱れず美しい描画を提供できる。また、SVGではアニメーション機能などもサポートしており、加えてJavaScriptなどを用いて動的なコンテンツの作成も可能となっている。

FXで実現したサマリグラフを図9に示す。

従来の専用監視盤と同等の表現力をブラウザ上の画面で実現している。このサマリグラフを描く作業はSVGを出力可能なグラフィックソフトで描き、サマリグラフをエンジニアリングするツールで動画のエンジニアリングすることにより、表示が可能となっている。

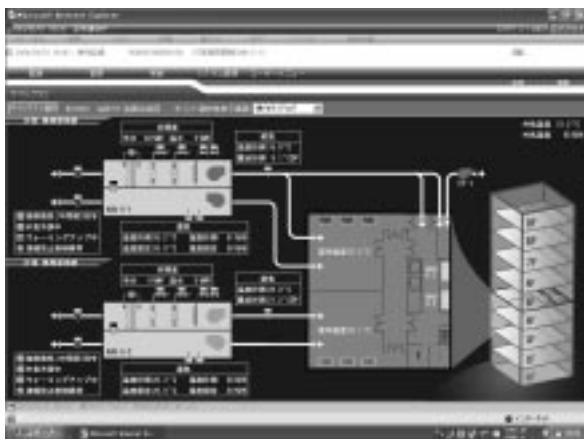


図9. FXのサマリグラフ表示

図10にSVGを利用した最大の特徴であるサマリグラフの任意倍率の拡大縮小した場合の表示を示す。



図10. FXのサマリグラフ拡大表示

この拡大表示機能はブラウザのプラグインとして提供されているSVG Viewerの機能を利用し実現されており、任意の場所の拡大、縮小が任意の倍率で可能となっている。

図11にFXサマリグラフ表示の仕組みについて示す。FXのサマリグラフはブラウザ上のJava Applet, Java Script, SVG

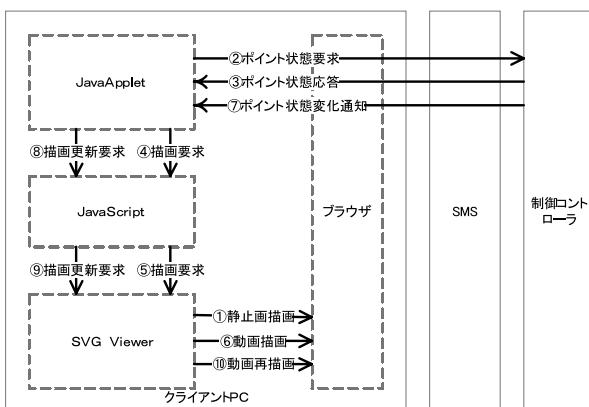


図11. FXサマリグラフ表示の仕組み

Viewerプラグインを利用して実現している。

監視画面はユーザーからサマリグラフの画面表示要求を受け付けると、ブラウザにサマリグラフ画面を表示し、SVG ViewerがSVGファイルを読み込み、静止画の表示をおこなう。(図11①)

JavaAppletは起動後、制御コントローラにポイントの情報を取得し(図11②,③)、取得した情報から、動画の色情報、形状情報、数値情報などを生成し、JavaScriptに通知をおこなう。(図11④)

JavaScriptはJavaAppletとSVG Viewerとの仲介を担当し、JavaAppletから通知された動画の情報をDOM(Document Object Model)を介してSVG情報を書き換える(図11⑤)。SVG ViewerはSVG情報をブラウザ上への画面表示をおこなっている。

#### 4. フレキシブル設計

FXのJava Appletを使用して実現している画面では、初期画面作成、画面の海外言語対応、将来の機能追加を容易にするためにFXフレームワークという仕組みに沿って画面を作成している。

まずFXフレームワークの概要を説明する。

FXフレームワークはフレームワーク本体、Javaのプログラムで作成された部品と画面ごとに定義される3つのXMLファイルから成る。

フレームワークはユーザーからの画面表示要求を受け付けると、指定された画面の部品パラメータXMLファイルを読み込み、XMLで指定された部品をXMLで指定されたパラメータを設定しつつ生成する(図12)。部品パラメータのうち文字情報は部品リソースXMLファイルに記述されており、この文字パラメータも生成時に部品に設定される。

部品は生成時の動作、イベント受信時の動作が決められており、生成時に渡されるパラメータにより動作が変更できるようになっている。

生成された部品は画面上に表示され、ユーザーや通信からの入力およびフレームワークからのイベント通知を待つ。

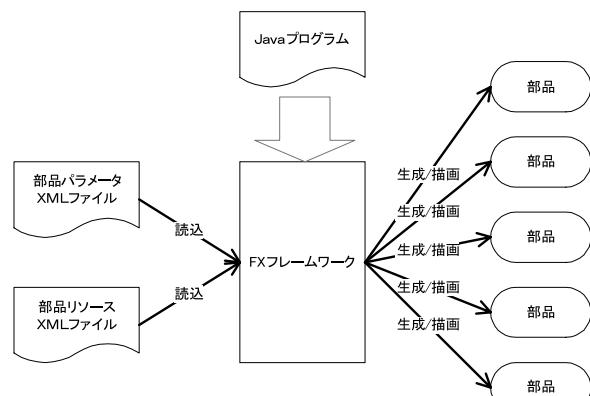


図12. FXフレームワーク部品生成の仕組み

図13は通信による状態変化通知やユーザーからの入力が発生した場合の流れを示している。部品は入力を受け取ると、フレームワークへと通知をおこなう。

フレームワークは部品からの通知を受け付けるとイベント定義XMLファイルを読み込み、XML上に記述されたイベントを各部

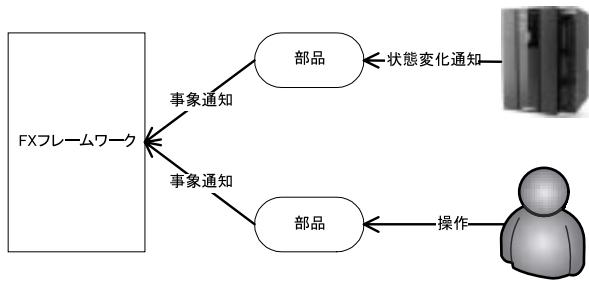


図13. FXフレームワーク事象通知の仕組み

品に通知する(図14)。

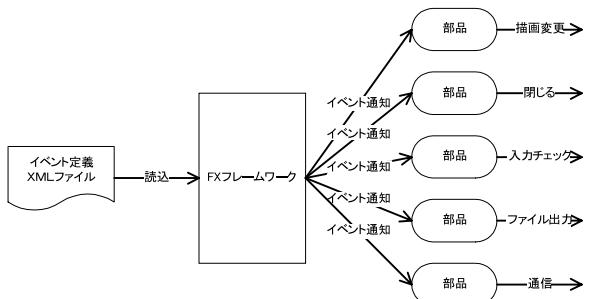


図14. FXフレームワークイベント通知の仕組み

部品はイベントが通知されると,Javaプログラムに記述された動作をおこない,画面描画の更新,ダイアログを閉じる,ユーザーからの入力をチェックする,ファイルを出力する,通信を行うなどの動作をおこなう。このようにして,動的に画面を変化させることを可能にしている。

このようなフレームワークの仕組みは海外言語対応やカスタマイズ対応が容易になるように設計され作成されている。

以下にこれらの対応をどのように行っているか仕組みを説明する。

#### 4.1. 海外言語対応

savic-netFXは現在,日本語,英語,韓国語,中国語,台湾語の5カ国語に対応している。

従来システムでの監視画面の海外言語対応では,画面に表示する文字の変更および文字の長さの変更による画面レイアウトの変更が海外言語対応の作業であった。

そこでFXではこれらの作業で変更する文字情報や画面レイアウトの情報を画面プログラムと分離し,部品パラメータXML,部品リソースXMLファイルとして定義した。そして,このXMLファイルを編集可能な用語入力ツールおよび画面作成ツールを開発した。これらのツールで作成したXMLファイルをサーバに言語ごとに場所を分けて配置し,クライアントの言語に応じて読み込むXMLファイルを変更することにより,1台のSMSで複数の言語の画面を表示することを可能としている。

図15に新しい言語の画面文字情報を作成するためのツールである用語入力ツールを示す。

このツールは翻訳元となる言語と比較しながら,目的の言語を投入し部品リソースXMLファイルを編集できる。

図16に画面のレイアウトを変更することが可能なツールである,画面作成ツールを示す。

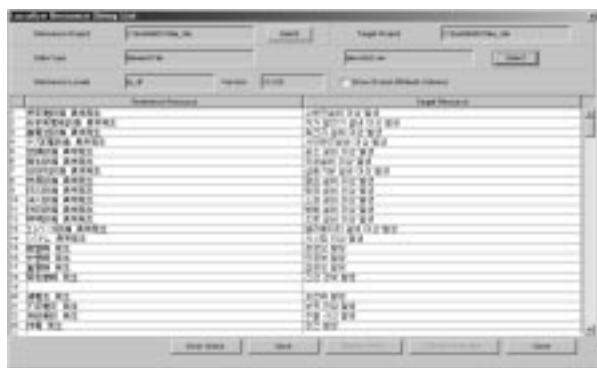


図15. 用語入力ツール

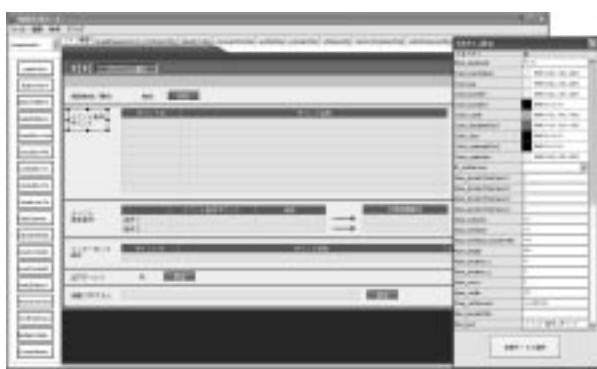


図16. 画面作成ツール

画面作成ツールは実際に表示される画面の状態を見ながら,レイアウトの変更や色情報の変更,フォントの変更,フォントサイズの変更,イメージの変更など様々な部品のパラメータの変更が可能となっており,修正した結果を部品パラメータXMLファイルとして出力することが可能となっている。用語入力ツールと画面作成ツールを利用して,全ての画面のXMLファイルを修正することにより,新しい言語への対応が可能となっている。

#### 4.2. 将来機能への対応性

FXではFXフレームワークの仕組みを利用し画面を作成することにより,存在する部品を組み合わせて新しい画面を作成することが可能となり,部品が増加するにつれて,画面の作成を効率よく行う事が可能となっている。

例えば,タイムプログラム一覧画面,力率トレンドグラフ画面という画面を開発した時,オリジナル画面であるタイムプログラム画面,トレンドグラフ画面に比べ機能追加がなされ,仕様は複雑になり,画面レイアウトも変更されているが,従来の画面に比べ半分程度の期間で作成できた。また開発工数の大部分は,新規機能に対する部品の作成および新規開発画面の試験時間であり,オリジナル画面の開発で作成した部品が有効活用されている。このように,FXは今後新たに要求される機能に対しても容易に対応する事が可能なしくみを内在している。

### 5. 印刷機能

従来の中央監視装置での印刷は,アプリケーションからプリンタへ直接印刷していた。FXでは印刷を行う際に一度PDF形

式で出力し、その後PDFファイルを印刷する形式をとっている。PDF形式は電子情報をやり取りするためのデファクトスタンダードとなっており、Adobe Readerを使用することにより、出力されたフォーマットのまま印刷、閲覧が可能である。

一度PDF形式で出力することにより、正確な印刷結果を一度画面上に表示して確認することが可能であるほか、印刷して紙として情報を保管するのではなく、PDFファイルとして保管することにより劣化することなくビルの運用情報を保管できるようになった。

図17に日月年報画面の印刷機能でPDFファイル出力を行ない、Adobe Readerで読み込んだ状態を示す。



図17. 日月年報のPDF出力

従来フォーマット付きの印刷は紙への出力のみに限られていたが、このPDF出力機能により紙からファイルでの蓄積を可能とした。

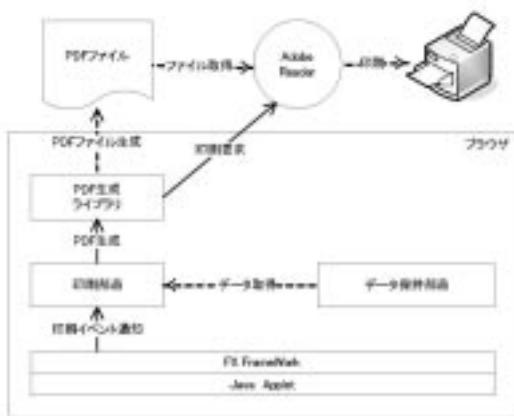


図18. FX印刷の仕組み

図18にFXで印刷を行った場合の流れを示す。

フレームワークから印刷要求を受信した印刷部品は印刷データを保持している部品からデータを取得し、PDFファイルを生成する。

このPDFファイルを生成する処理はクライアントPCのJava上で動作するように設計しており、サーバ側はデータの提供のみを行っている。そのため、複数クライアントから同時に印刷を行ってもサーバ側には大きな負荷がかからず、リアルタイム処理など定常処理に支障をきたさないようになっている。

## 6. おわりに

savic-netFXではBAシステムの監視機能をWeb上で実現することにより、汎用PCをネットワークに接続するだけで簡単に遠隔からの監視が可能になった。また同時に、従来のWeb監視装置の機能や性能を劣化させることなく、Webならではのユーザーインターフェースを追加することにより、さらに使いやすい監視装置として完成している。今後、FXはFlexibleのコンセプトを生かしながら、さらに機能を追加し成長していく予定である。

### 商標

savic-netFXは、株式会社山武の登録商標です。

JavaおよびJava関連の商品は米国Sun Microsystems, Inc. の米国およびその他の国における登録商標です。

Adobe、Adobeロゴ、Reader、SVG Viewerは、Adobe Systems Incorporated(アドビシステムズ社)の米国ならびに他の国における商標または商標登録です。

### 著者所属

山本 博之 ビルシステムカンパニー  
開発1部ソフトウェアグループ