

産業用監視制御システムの Web 化の検討

古城 仁士[†]東芝 ソフトウェア技術センター[†]

SVGを初めとした表示に関するWeb技術の普及と、クラウドコンピューティングの発展に伴い、電力制御システムなどの高い信頼性、応答性を求められる産業用監視制御システム（SCADA）をWeb化する試みがなされている。

その一方で電力制御システムはその特性上、各機器において発生した警告などの重要な情報をその発生から所定の時間以内に表示しなければならない。また多くの機器情報を俯瞰できる必要性から、単位時間当たりの通信量や描画負荷、処理量が多い傾向がある。また、ユーザー体験の関係上画面遷移を多用できないため、通常のWebページとは異なる困難さがある。

SCADAにおけるWeb化の利点としては以下が考えられる。

- A) 専用のライブラリを用いて開発されている既存システムとの比較において、一般的な技術を多用する事で情報の入手手段が多くなり、学習コストを低減可能となる。
- B) ブラウザは今日の多くの端末において標準的に搭載されており、それら端末においてSCADAを操作することが可能となる。これにより、実作業員と制御室のオペレータとの間で円滑な意思疎通が可能となる。
- C) SVG、CSSアニメーションをはじめとする高機能な表現手段を使用することが可能。特に凝った表現を用いない限りこれらを活用することで工数を維持しつつ表現能力上昇を実現可能となる。
- D) 機能面に限らず、デザインの面においても他の技術と比較して多くのOSS・情報・ツールが存在しており、これらを活用することで工数の削減が期待される。

一方でWeb化により発生する制限事項も多く、以下が考えられる。

- E) JavaScriptを使用したファイルへのアクセス手段が限られている。このためExcelなどの業務アプリケーションとの連携が困難。ファイルをサーバ側に転送し、サーバ側にて解析するなどの必要がある。
- F) JavaScriptによるバイナリ列の取り扱いが困難な場合がある。IE8やIE9への対応を考える場合、特にこれは問題となる。このためUTF-8によるファイル保存時にBOMを追加する場合などがクライアント側では完結できない。
- G) フォーカスや選択の可否、操作の可否などの設定が非Webアプリケーションと比較して一貫性がなく整備されていない。
- H) 単純なHTTPベースの通信ではSCADAに求められる警告通知を所定の時間内に完了する事が困難。
- I) HTTPベースの通信の場合、TCP/IPと比較して通信量が多くなる傾向がある。SCADAの場合、常時多くの情報を交換しているためこの傾向が顕著となる。発展途上国においては通信インフラが整備されていない場合がある。
- J) 画面遷移を主体としないWebアプリケーションの形式にて実装される場合、擬似的な画面遷移や、ウィンドウサイズへの対応など、通常のWebページと異なる部分が多く、実装方法が一貫していない。
- K) SLDやGISにおいてSVGを使用した場合、その規模によってはパフォーマンス上の問題が発生する。
- L) その他、クリップボードとの連携やマウスキャプチャの問題など。

これらの問題のいくつかについてはHTML5および関連技術によって改善が図られている。本発表ではこれらをいかに改善するかについての試みについて述べる。

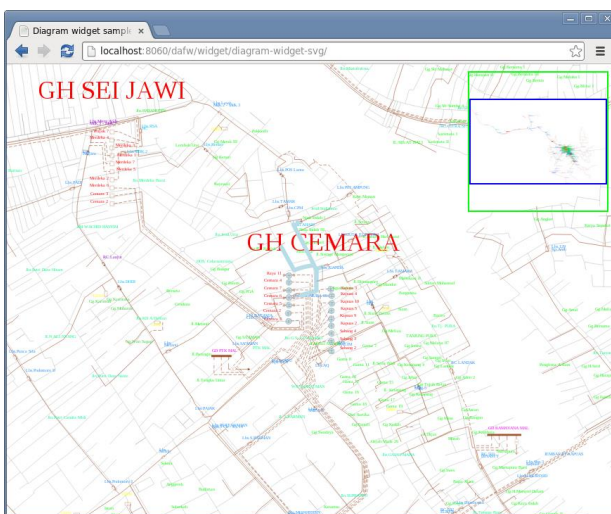
Studies on web-based SCADA

[†]Masashi Kojo

Software Technology Center
Toshiba Corporation

例えば、通信関係においてはWeb Socketにより双方向通信が可能な場合がある。データベース側での（またはCEPによる）状態検知が適切に行われれば、状態を通常求められる1秒以下に抑えることが出来る。IE8またはIE9においてはWeb Socketを使用することはできないためその場合はCometに切り替える。この場合、通信量が犠牲となり、（実測で8倍程度の）転送量増となる。日本国内においてはこの転送量増は大きな問題とならないが通信環境の良くない地域においては問題となる。また、Cometの場合はセッション管理も問題となる。通常セッションはCookieなどで管理されるが、このCookieは同じWebアプリケーションの場合同じであるので、サーバ側からみて同一のクライアントにしか見えない。このため、セッションIDに加えてそれぞれを識別するIDをさらに付与する必要が出る。しかしながらそれでもなお、同一Cookieで多並列にてHTTP通信が行われることを避けられないため、既に無効化されたCookieが使用されることを抑制する手段がないなどの問題が出る。

切なFPSは得られない。新しく標準化されたWebGLを使用し、全ての描画を一つのVBOに詰め、VSにて色替えやアニメーションを行うことで60FPSを達成できる。ただし、この場合、マウスとのヒットテストなどを実装しなければならない。



また他の例としてはWebGLがある。大規模なSLDやGISをSVGにより表現した場合、Chromeにおいてもそのパフォーマンスはテストケースにおいて数FPS程度しか出せない。Google Map等においては画像化された地図を用い、高いFPSを得ているが、SLDやGISにおいてはほぼ全ての送電線などの線がアニメーション対象であるためこの方法を取ることはできない。SVGが高いパフォーマンスを得られないのはその描画方法が故と考えられるが、SVGを分割するなどしても適