

オープンソースを活用したWebベースGUI開発環境

Web-Base GUI Development Environment Using OSS

中島 暢康

清水 伸夫

■ NAKAJIMA Nobuyasu

■ SHIMIZU Nobuo

一般に、GUI (グラフィカルユーザーインターフェース) は動作する機器のOS (オペレーティングシステム) などの基本システムに強く依存しているため、パソコン (PC) やスマートフォンといった基本システムが異なる機器で用いる場合には、機器ごとに専用のGUIを開発する必要があった。

東芝は、様々な機器を表示端末として用いる社会インフラシステム向けGUIを開発するために、WebブラウザをベースとしたGUI開発環境を構築した。このWebベースGUI開発環境では今回、グラフィカルに部品を配置できるツールと、オープンな標準規格であるHTML5 (Hypertext Markup Language 5) を用いて社会インフラシステム特有の図面をわかりやすいグラフィックスで表示できるモジュールを、それぞれOSS (オープンソースソフトウェア) を用いて開発した。Webを用いることで、様々な機器のGUIを統一的に開発できる環境が整った。

A graphical user interface (GUI) is closely dependent on the operating system of the device in which it is installed. Individual GUIs therefore have to be developed for different devices such as PCs and smartphones.

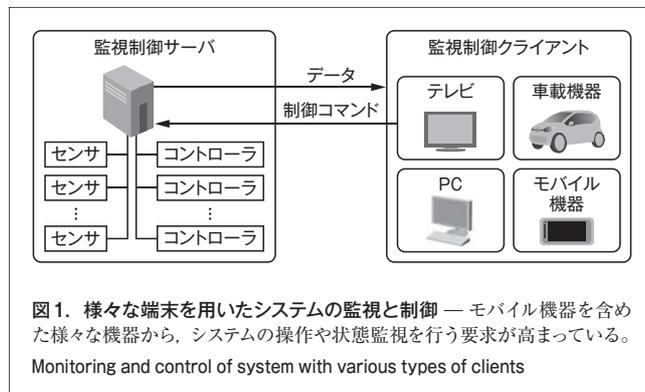
Toshiba has developed a Web-based GUI development environment to provide a common GUI to devices with Web browsers used in social infrastructure systems. In order to realize this environment, we have developed the following technologies utilizing open source software (OSS): (1) tools for the graphical deployment of GUI parts, and (2) modules for the creation of drawings characteristic to the social infrastructure system using HTML5 (HTML: Hypertext Markup Language). These technologies make it possible to develop Web-based GUIs using a unified routine.

1 まえがき

GUIは、PCはもとより、組み込み機器や産業用システムでも一般に利用されており、システムの操作や状態監視を行う際の入出力手段として欠かせないものになっている。

一方で、スマートフォンやタブレットなどのモバイル機器におけるグラフィックス性能の進歩は目覚ましく、PCと比べても遜色のない表現が可能である。このため、可搬性の高いモバイル機器を用いて、PCと同様の入出力方法により、システムの操作や状態監視をいつでもどこでも可能にしたいという要求がある。また、車のダッシュボードや車載モニタに周辺の交通状況を表示させたり、テレビに家庭内や地域の電力使用量を表示させたりすることも求められている (図1)。これらを実現するためには、現在のPCと同様のGUIを様々な機器に実装することが必要である。しかし従来のGUI実装は、OSや開発言語などに強く依存しているため、通常、システムごとにGUIの開発が必要になる。このため、様々な機器をGUI端末として用いるには、開発に多くの工数が掛かるという課題があった。

この課題を解決するには、スマートフォンやタブレットでは既に高機能なWebブラウザを備えていることから、アプリケーションをWebブラウザ上で動作させるWebアプリケーションにすることが考えられる。Webブラウザを用いると様々な機器の



GUI開発を統一的に行うことができ、少ない工数で複数の機器をGUIクライアントとして用いることができるようになる。

東芝は、このようなWebアプリケーションのGUIを効率よく開発するために、WebベースGUI開発環境を構築した。ここでは、開発したWebベースGUI開発環境の特長について述べる。

2 WebベースGUI開発のメリット

2.1 Webを用いた統一的なGUI開発

スマートフォンやタブレットにも、PCと同様の機能や性能を

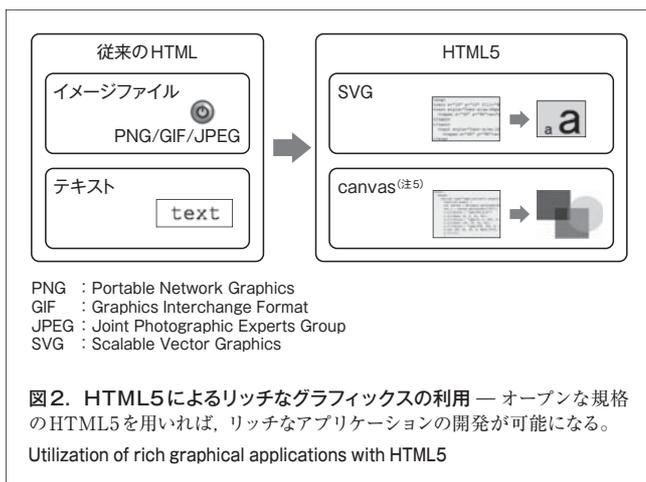
持ったWebブラウザが搭載されてきていることから、WebベースGUIを用いて、様々な機器向けのGUIを統一的に開発することで、GUI開発工数を削減できる。これは、WebブラウザをGUIの実行環境とし、HTMLや、JavaScript^(注1)、CSS (Cascading Style Sheets)などのWeb標準技術を用いれば、Webブラウザをサポートする様々な機器で同じようにGUIを動作させることができるからである。

更にWebアプリケーションでは、既存のWebサービスで提供されるデータや機能を組み合わせて、新しいサービスを容易に作るができるという利点もある。例えば地図情報サービスでは、表示したい画像などのオブジェクトを緯度と経度の情報とともに与えると、地図上にそのオブジェクトを重ね合わせて表示するインタフェースが提供されている。これにより、地図上にグラフを配置して、データを表示させるGUIを簡単に実現できる。

2.2 HTML5を利用したリッチなGUI

Webでは、リッチインターネットアプリケーション (RIA) と呼ばれる操作性や表現力に優れたアプリケーションが提供されてきたため、このようなリッチなGUIは今後も必要である。

従来のRIAは、Webブラウザ内の操作性や表現力を強化するために、Adobe[®](注2) Flash[®](注3)やJava^(注4)のようなWebブラウザには同梱されていないプログラムをプラグインとして用いて実現するものであった。Adobe[®] Flash[®]やJavaのプラグインはOSやWebブラウザの違いを吸収しているため、それらに対応したアプリケーションを何種類も開発する必要はなかった。一方でこのようなプラグインは、ユーザーによるインストールが必要であることから、ユーザーに対して操作を強いること



(注1) JavaScriptは、Oracle Corporation又は関係会社の米国及びその他の国における商標又は登録商標。

(注2)、(注3) Adobe, Adobe Flashは、Adobe Systems Inc.の米国及びその他の国における登録商標又は商標。

(注4) Javaは、Oracle Corporationあるいはその関連会社の登録商標。

(注5) プラグインを使わずに、JavaScriptベースでWebブラウザ上に図を描くために策定された仕様。

になる。更に近年、このようなプラグインをサポートしないOSも普及し始めていることから、従来のようなプラグインで統一的なGUIを開発することができなくなっている。

これに対してWebでは、現在、新しいオープンな規格としてHTML5の標準化が進められている。HTML5ではリッチなグラフィックス機能が標準化されていることから、これを用いれば特定のソフトウェアに頼ることなくリッチなアプリケーション開発ができるという利点がある(図2)。これらのグラフィックス機能は、既に主要なブラウザで実装され、サイトでの利用も始まっている。

3 WebベースGUI開発環境の特長

2章で述べたように、GUIアプリケーションをHTML5を活用したWebベースで開発することで、様々な機器を端末として用いる表現力に優れたアプリケーションを、統一的に開発することができる。一方、GUIアプリケーション開発では、効率的な実装が求められる。

そこで当社は、WebベースでGUIアプリケーション開発を支援するWebベースGUI開発環境を構築した。GUIアプリケーション開発では、プログラムの作成とそのできばえを画面上で確認する作業を繰り返し行う必要があるため、効率的な画面の実装手段が求められる。また、ボタンのようなGUIを構成する基本要素だけでなく、グラフや対象システム特有の形式によるグラフィカルな情報の表示を、Web技術で実現する。そして、それらを部品化することで再利用可能にし、画面への実装を効率化する必要がある。このWebベースGUI開発環境では、OSSを活用することで、グラフィカルな部品配置や、情報をわかりやすく表示する手法を実現し、Webベースアプリケーション開発の効率化を図っている。ここでは、GUI開発環境の構成、及びGUI開発の効率化手法について述べる。

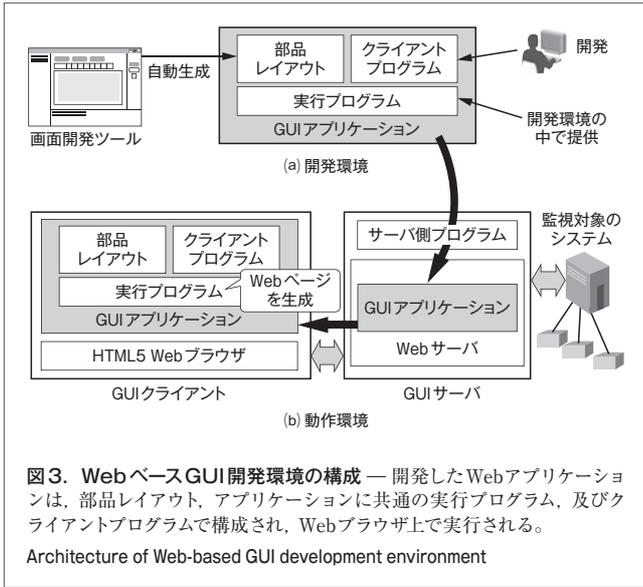
3.1 WebベースGUI開発環境の構成

開発環境と開発したWebアプリケーション動作環境の構成を、図3に示す。

Webアプリケーションは、一般にクライアントとサーバから構成される。ユーザー側のGUIクライアントはWebブラウザ上に実装され、ユーザーからの入力受付や処理結果の表示と、サーバ側からの情報の表示を行う。一方サーバ側は、クライアント上に表示すべきデータを収集し、データベースなどを用いて管理するとともに、収集したデータをクライアント側に送信する役割を持っている。GUI開発環境は、このようなWebアプリケーションのうちクライアント側の開発を行うためのものである。

3.2 WebベースGUI開発の効率化

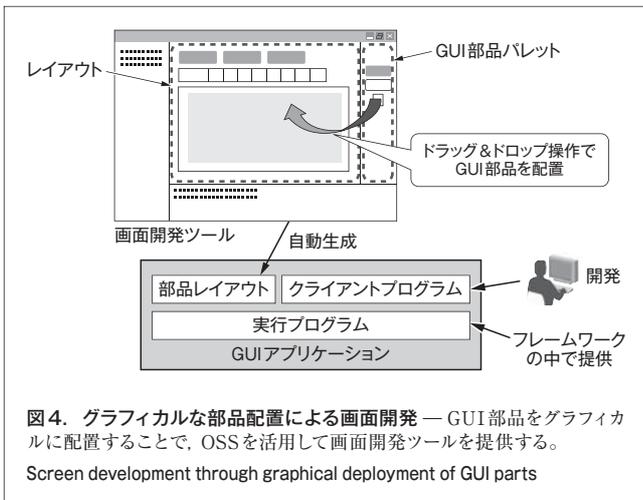
3.2.1 グラフィカルな部品配置 GUI開発では、画面レイアウトの開発をわかりやすく簡単に行うことが求められ



る。例えば Visual Studio[®](注6)のような統合開発環境では、ボタンなどのGUI部品をグラフィカルに配置することで画面レイアウトを開発できるため、画面レイアウトの実装を簡単に行うことができる。このため、Webアプリケーション開発における画面レイアウトも、Visual Studio[®]と同様の手法で開発できることが望まれる。

これに対して、今回のGUI開発環境では、OSSを導入してグラフィカルな画面レイアウトができるような画面開発ツールを構築し、部品のドラッグ&ドロップ操作によって画面のレイアウトを開発可能にした(図4)。

この画面開発ツールを用いたGUIアプリケーションは、画面開発ツールによって自動生成される部品レイアウト、アプリケーションに共通の実行プログラム、及びそれぞれの画面によって異なる処理を記述するクライアントプログラムの三つの



(注6) Visual Studioは、米国Microsoft Corporationの米国及びその他の国における商標又は登録商標。

要素から構成される。このうち実行プログラムは、ボタンやテキスト入力領域といった基本的なGUI要素をGUI部品として提供するとともに、GUI部品の配置を定義する部品レイアウトに従ってWebページを生成する。また、この実行プログラムはサーバとの通信を行うためのインタフェースを提供しており、クライアントプログラムはこのインタフェースを利用してユーザーからの入力をサーバ側に通知したり、サーバ側からの情報を受け取ったりする。これら三つの要素から成るGUIアプリケーションはWebサーバ上に配置され、実行時にGUIクライアントであるWebブラウザにダウンロードされ、実行される。

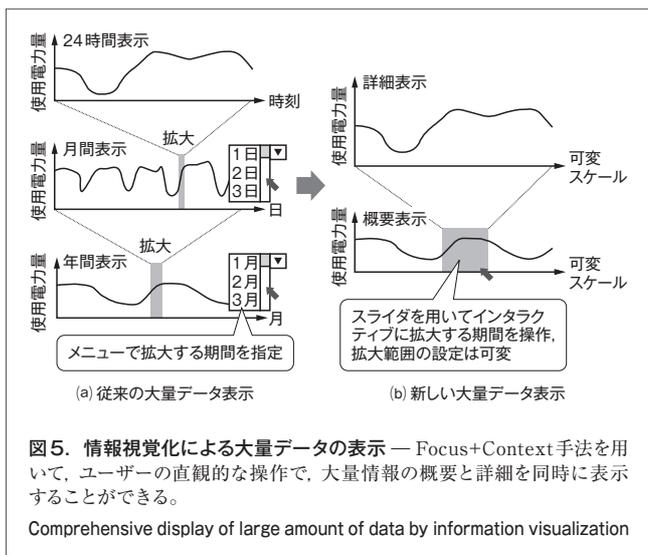
3.2.2 HTML5によるグラフィカルな図面表示 今回開発したWebベースGUI開発環境では、ボタンのような基本的なGUI部品に加えて、よりグラフィカルな図面をサポートすることが要求される。例えば電力系統の監視制御システムでは、単線結線図と呼ばれる図面を用いてシステム内の機器の数値情報や障害の有無などを表示しているが、このような図面をWebブラウザ上でも表示する必要がある。そこで、このような複雑でグラフィカルな図面を表示するために、HTML5でWeb標準グラフィックスとしてサポートされているSVG (Scalable Vector Graphics) を用いた。

SVGは、スクリプトによって画面上の線の色を変更したり、表示する文字を変更したりできるため、サーバ側からは色や文字の情報だけを送信すれば、送られたデータをクライアント側のプログラムで表示できる。これによって、情報が埋め込まれた図面をサーバ側で生成してクライアントに送信することなく、動的にクライアント側での表示処理ができるため、サーバからクライアントに送るデータ量が抑えられるという効果もある。

3.2.3 情報視覚化手法を用いた大量情報の表示 大量の情報をGUIを用いて表示する場合、限られたサイズの画面上にわかりやすく表示することが求められる。そこで、リッチなグラフィックスとマウスなどの入力手段を組み合わせ、ユーザーの直観的な操作で大量の情報をわかりやすく表示できる情報視覚化手法を適用し、OSSのライブラリを用いてWeb上への実装を実現した⁽¹⁾。

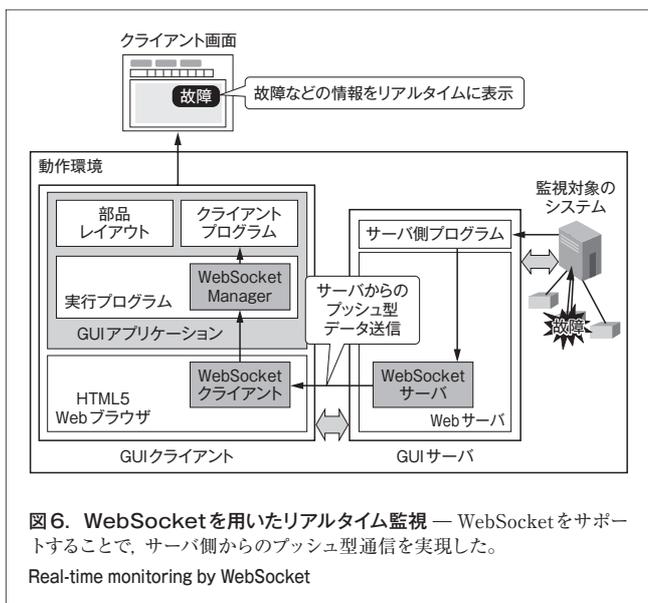
使用電力量データの表示例を図5に示す。従来、使用電力量の日内変動、月内変動、あるいは年間を通した変動は、それぞれ別々のグラフを用いて表示していた。これに対して、情報視覚化手法の一つであるFocus+Contextを用いると、データの概要と詳細を同時に表示できるため、ユーザーはデータ全体の傾向と詳細を一目で理解することができる。このように大量のデータをわかりやすく表示する手法を今回のGUI開発環境でも利用できるようにするため、現在、GUI部品化を進めている。

3.2.4 WebSocketによるリアルタイム通信 社会インフラシステムでは、GUIを用いてシステムの状態変化をリアルタイムに表示することが求められる。しかし現在のWeb



技術では、サーバ側からデータを得るにはクライアントからのリクエストが必要なプル型通信しかなく、リアルタイム性に欠けるという課題がある。これに対して、サーバ側からのプッシュ型通信を実現する新しい通信プロトコルとしてWebSocket^(注7)が策定されており⁽²⁾、今回のGUI開発環境ではこのWebSocketをサポートした。

WebSocketは、他のWeb通信方式と同様にWebブラウザからAPI (Application Programming Interface) が提供される。そこでこのAPIを隠蔽して、従来の通信方式と同様な書式のAPIを新たに提供することで、従来と同様のプログラミングができるようにしている。これにより、リアルタイム性を確保した通信方式をわかりやすく簡単に利用できるようにした(図6)。



(注7) HTTP (Hypertext Transfer Protocol)と違い、クライアントからのリクエストなしでサーバ側からデータを送信できる双方向通信技術。

4 あとがき

今回、HTML5を用いたWebベースGUI開発環境を構築した。この開発環境では、GUIクライアントをWebブラウザ上で動作させていることから、Webブラウザを備えている様々な機器で、共通に動作するGUIクライアント開発が可能になった。また、HTML5で導入されているWeb標準のグラフィックス形式を用いて、複雑な図面を表示するGUI部品を開発した。これを開発環境に組み込むことで、GUI開発が効率よく行えるようになった。

今回作成したGUIは、例えばタブレットのブラウザでも同じように動作することを確認しており、この開発環境を用いることで様々な機器に共通なGUIを提供できることを検証した。

現在、このWebベースGUI開発環境では、色や形といったスタイルに関する定義と動作ロジックの定義を分離して開発できるGUI部品を開発中である。これにより、画面イメージとアプリケーションロジックの独立性が高まり、互いに影響されずにそれぞれの修正や拡張作業ができることから、拡張性や保守性が大幅に向上することが期待される。今後、このようなGUI部品を充実させることで、GUI開発環境の更なる機能強化を図っていく。

文献

- 梁 連秀. 情報視覚化の電力管理システムへの適用. 東芝レビュー. 66, 12, 2011, p.56-57.
- Fette, I. et al. The WebSocket Protocol. Internet Engineering Task Force (IETF), 2011, 70p., (RFC 6455). <<http://tools.ietf.org/html/rfc6455>>, (accessed 2012-07-16).



中島 暢康 NAKAJIMA Nobuyasu

ソフトウェア技術センター 先端ソフトウェア開発担当参事。
ユーザーインタフェースソフトウェア技術の開発に従事。電子情報通信学会, IEEE会員。
Corporate Software Engineering Center



清水 伸夫 SHIMIZU Nobuo

ソフトウェア技術センター 先端ソフトウェア開発担当主幹。
GUIなどミドルウェアの開発に従事。情報処理学会会員。
Corporate Software Engineering Center