

Webブラウザエンジン WebKitの映像製品への適用

Application of WebKit Web Browser Engine to Audiovisual Devices

平野 裕 深井 祐介

■HIRANO Yutaka

■FUKAI Yusuke

近年、WebサービスやHTML5 (Hypertext Markup Language 5) 及びその関連規格が普及し、Webブラウザ上で高機能なアプリケーションが開発できるようになった。しかしWebブラウザは複雑なソフトウェアであり、デジタルテレビ (TV) をはじめとする映像製品、いわゆる組込み機器に搭載して十分に活用することが難しい。

そこで東芝は、デジタルTVなどへの適用を目指し、OSS (オープンソースソフトウェア) のWebブラウザエンジンである“WebKit”をもとに、映像製品上で軽快かつ安全に動作するWebブラウザの研究開発を行っている。また、映像製品のGUI (グラフィカルユーザーインターフェイス) を実現するプラットフォームとしてWebKitを活用する試作を行っており、開発効率と移植性の大幅な向上を目指している。

Accompanying the diffusion of Web services and HTML5 (HTML: Hypertext Markup Language) with related standards, Web browsers have become able to provide greater functionality to application developers. Due to the complexity of Web browser software, however, it is difficult to install and utilize in audiovisual (AV) devices.

To overcome this issue and realize the application of Web browsers as a GUI platform for digital TVs and other devices, Toshiba is developing a Web browser that operates lightly and safely on AV devices based on the WebKit open source software for Web browser engines. We are also conducting a trial aimed at utilizing WebKit as a graphical user interface (GUI) platform for AV devices, with the goal of realizing major improvements in development efficiency and portability.

1 まえがき

Web上のドキュメントを記述するための言語であるHTMLの現行規格は、1997年に定められたHTML4.0及びそれをマイナーバージョンアップしたHTML4.01である。現在、その後継としてHTML5⁽¹⁾の策定が進められており、Webブラウザへの実装も行われている。

HTML5及びその関連規格^(注1)では様々な新機能が導入され、例えば次のような機能が提供される。

- (1) ドキュメントの意味づけの、より詳細な指定
- (2) ファイルやストレージなど、ローカルリソースにアクセスするための標準的な手段
- (3) ビデオやオーディオの要素と、その再生を制御する標準的な手段

これらの新機能により、いっそう高機能なアプリケーションがWebブラウザ上で開発できるようになる。

また近年、Webサービスが広く普及し、Webサービスとの連携も含めたアプリケーションプラットフォームとしてのWebブラウザの重要性が増している。

(注1) HTML5は、HTML4.01の直接の後継規格に加えて、ファイル操作や、データベース操作、位置情報取得などの関連規格も含めることがある。ここでは、関連規格も含めてHTML5と表記する。

しかし、デジタルTVをはじめとする映像製品では、メモリ量や、CPU能力、プロセスモデルなどの制限により、最新のHTML5対応ブラウザを導入してその機能を活用することが難しい。

そこで東芝は、デジタルTVなどへの適用を目指し、OSS (オープンソースソフトウェア) のWebブラウザエンジンであるWebKit⁽²⁾をもとに、映像製品、すなわち組込み機器上で軽快かつ安全に動作するWebブラウザの研究開発を行っている。更に、映像製品のGUIをWebブラウザ上で実現することで、Webと映像が連動したアプリケーションを効率的に実現できるソフトウェアプラットフォームの構築も行っている。

ここでは、WebKitの概要、及び当社におけるWebブラウザ活用の取組みについて述べる。

2 WebKitの概要

2.1 WebKitとは

WebKitとは、オープンソースのWebブラウザエンジンである。Webブラウザエンジンとは、単独ではWebブラウザとして動作せず、様々なライブラリやアプリケーションコードと組み合わせることでWebブラウザとして動作するソフトウェアである。WebKitを利用した主なWebブラウザには、Safari^(注2)や、

Google Chrome™^(注3)、Android™^(注4)の標準ブラウザなどがあ

る。WebKitはオープンソースプロジェクトであるためボランティアの開発者も参加しているが、開発の中心になっているのは、WebKitを利用したブラウザやシステムのベンダーに雇用された開発者である。開発に必要なインフラも、それらのベンダー企業から提供されている。

2.2 ポートとアプリケーション

WebKitは様々な環境に移植されている。プラットフォームごとの、実装された結果としてのソフトウェア（以下、実装と言う）をWebKitでは“ポート”と呼び、macや、win、chromium、gtk、qtなどの名前が付けられたポートがある。

また、WebKitはネットワークアクセスや、スレッド^(注5)管理、GUI描画などのためにライブラリを利用するが、利用するライブラリはポートごとに異なる。このため、WebKitのコア機能からライブラリを利用するための実装もポートごとに異なる。この、ポートごとに異なる実装を、以下では“プラットフォーム依存コード”と呼ぶ。

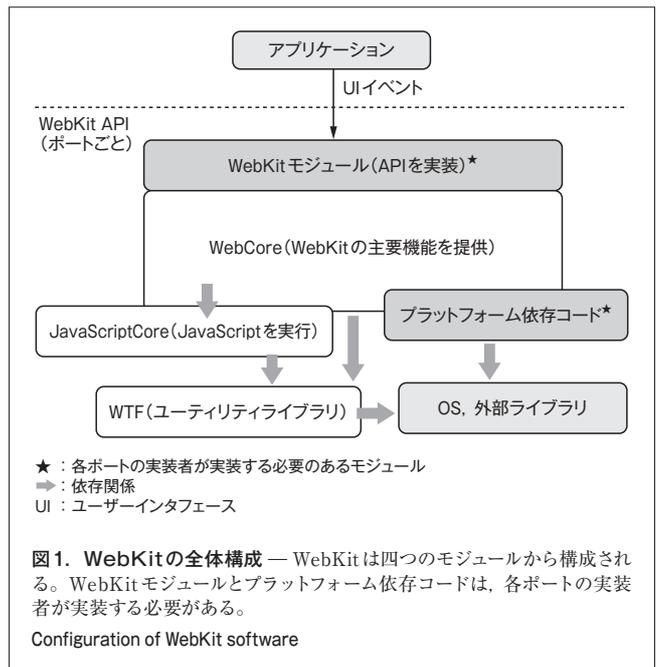
更にポートでは、アプリケーション用のインタフェース（API: Application Programming Interface）を用意する必要がある。アプリケーションは、このAPIを通してWebKitの機能を利用する。APIの設計は、ポート開発者の責務であり、ある程度の概念的な共通点はあるものの、ポートごとに独自のものである。

2.3 WebKitの内部構成

WebKitは、WebCoreモジュール、JavaScriptCoreモジュール、WebKitモジュール、及びWTFモジュールという四つのモジュールから構成される（図1）。WebブラウザエンジンとしてのWebKitとの混乱を避けるため、ここではモジュールとしてのWebKitを“WebKitモジュール”と呼ぶ。

WebCoreは、WebKitの主要な機能を提供するモジュールである。HTMLの解析や、レイアウト、レンダリング^(注6)、DOM（Document Object Mode）^(注7)API、WebSocket^(注8)API、プラグインAPIなどの機能が含まれる。

JavaScriptCoreは、JavaScript^(注9)の実行環境である。ただし、独自のJavaScript実行環境v8を持つchromiumポートのように、JavaScriptCoreの代わりに別の実行環境を用いる



こともできる。

WebKitモジュールは、APIを実装するものである。アプリケーションは、WebKitモジュールが提供するAPIを通して、WebCoreやJavaScriptCoreの機能を利用する。

WTFは、WebCoreやJavaScriptCoreで利用される、ハッシュテーブル^(注10)などのデータ構造やメモリ管理、スレッドなどの共通的な機能を提供するユーティリティライブラリである。

3 東芝におけるポート開発

3.1 ポート開発の方法

ポート開発には、WebKitのリポジトリ^(注11)上で開発を行う方法と、ある時点でのリポジトリのスナップショットを取得して、それを起点に開発する方法（ローカルブランチ）の2種類がある。リポジトリ上での開発には、次のような利点がある。

- (1) 最新のWebKitを常に利用できる。
- (2) 他者によりWebKitの変更が行われるとき、自身の変更が考慮される。
- (3) リポジトリや、バグ追跡システム、レビューなどの開発インフラを利用できる。

一方、ローカルブランチでの開発には、次のような利点がある。

- (1) 他者による変更で自身の環境が壊れる心配がない。

(注2) Safariは、Apple Inc.の商標。

(注3)、(注4) Google Chrome、Androidは、Google, Inc.の商標。

(注5) 並列処理を行うための、処理の単位。

(注6) 解析された文書情報から、画面に描画されるイメージを生成すること。

(注7) World Wide Web Consortium (W3C)が勧告している、HTML文書やXML (Extensible Markup Language)文書をアプリケーションから利用する機能。

(注8) サーバとクライアント間の双方向通信を実現するために、W3CとInternet Engineering Task Force (IETF)で策定が進められている通信プロトコル。

(注9) JavaScriptは、Oracle Corporation及びその子会社、関連会社の米国及びその他の国における登録商標。

(注10) キーワードとそれに対応する値の組を表として持ち、キーワードに対応する値をすばやく参照するためのデータ構造。

(注11) ソフトウェアのソースコードなどを一元的に管理するため、ソースコードやデータの仕様などの情報が保管されている、広い意味でのデータベース。

(2) 実装している機能を、リリース時まで公開しなくてよい^(注12)。

WebKitは活発に開発され、変更も頻繁に行われるため、ローカルブランチで開発する場合でも、一定期間ごとに最新のWebKitを取り込むほうがよい。ただし、その際には、WebKitリポジトリ上での変更と、ローカルブランチでの変更を統合する必要がある。

当社は、ローカルブランチでポートを開発する方法を採用した。つまり、次のサイクルを一定周期ごとに行っている。

- (1) WebKitリポジトリのあるスナップショットを取得
- (2) ローカルブランチを作成
- (3) 独自の機能を開発

ローカルブランチ方式を採用する場合、WebKitのソースコードを変更すると、最新のWebKitを反映する際に非常に手間が掛かる。このため、WebKitのソースコードをできるだけ変更しないように必要性を厳しく吟味したり、変更する場合にはファイルを分離したりして、WebKitのソースコードを最新のものに置き換える際の作業量を削減している。

3.2 組込み機器へ搭載するためのメモリ削減

WebKitは多数のライブラリを使用するので、メモリ量の少ない映像製品などの組込み機器に搭載することが難しい。できるだけライブラリの数やサイズを削減するため、当社は次のような工夫を行っている。

- (1) WebKitから、例えばSVG (Scalable Vector Graphics) 機能のような必要でない機能を取り除き、使用するライブラリ数を減らす。WebKitに搭載されたビルドスイッチ機能^(注13)を使うこともある。
- (2) ポートを構成する際に、できるだけ単体で動作するライブラリを選び、ライブラリが芋づる式に増えることがないようにする。
- (3) 各ライブラリは、WebKitと関係のない機能を削除して作成する。

3.3 プロセス分離機能の開発

デジタルTVなど映像製品では、パソコン(PC)とは異なり、一つの主要なプロセス(以下、システムプロセスと呼ぶ)の中で様々なアプリケーションが動作していることが多い。

一方、PC上のWebブラウザとして生まれたWebKitは、メモリなどのリソースが足りなくなると自分のプロセスを終了させるように設計されている。このため、WebKitを単に移植しただけでは、リソース確保に失敗したとき、クラッシュしてシステム全体を停止させてしまう。しかし、WebKitのソースコードを修

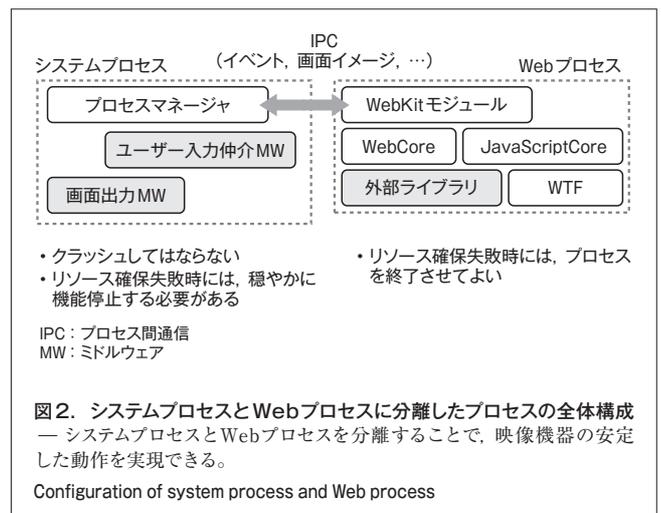
正してこの問題を解消しようとする、膨大な量の修正が必要になる^(注14)。

そこで当社は、Webブラウザのほとんどの部分を分離して別プロセス(以下、Webプロセスと呼ぶ)として動作させることにより、コードの変更量を最小限に抑えつつ、安定させることを図った。これにより、リソースの確保に失敗してプロセスが終了しても、システムプロセスは影響を受けない。

プロセスを分離する機能としては、WebKit2と呼ばれる、ポートに依存しない機能がWebKitに実装されている。また、chromiumポートでは、独自にプロセスを分離する機能が実装されている。ただし次の理由から、当社はこれらの機能を利用せずに独自のプロセス分離機能を開発した。

- (1) chromiumのプロセス分離機能は、タブごとにプロセスを作成するため、メモリ使用量が增大してしまう。
- (2) chromiumのプロセス分離機能は、chromium以外から使用できるよう設計されていない。
- (3) システムプロセス側のモジュールは、エラーが発生した際にも、TVの方式に従って安全に停止させなければならない。また、そのためには、全体が詳細に把握できるほど小さくしなければならない。WebKit2はこれらの条件を満たしていない。

当社が開発したプロセス分離機能は、Webプロセスと通信を行って画面の描画やユーザー操作イベントなどのメッセージの仲介を行うモジュール(プロセスマネージャ)を開発して、システムプロセスの中に実装するものである。このモジュールは、軽量かつ高速に動作し、またエラーが発生したときにも安全に処理を終了するように、当社が独自に開発した。システムプロセスとWebプロセスに分離したプロセスの全体構成を図2に示す。



(注12) WebKitのソースコードライセンスは、LGPL (GNU Lesser General Public License) やBSD (Berkeley Software Distribution) ライセンスなど多様である。LGPLのライセンスを受けたソースコードを編集した製品をリリースした場合には、ソースコードを開示する義務がある。

(注13) プログラム(この場合はWebKit)をコンパイル化するための設定。

(注14) メモリ確保関数の失敗時にエラー情報を送出するよう変更するだけでなく、それを呼び出す部分の多くで、失敗時のエラー処理を実装する必要がある。

4 WebKitを活用したGUI

HTML5規格により、従来のWebブラウザでは使用できなかった機能が標準的な操作で可能になる。当社は、映像製品のGUIをWebKitを用いて実現するための研究開発を行っている。これにより、従来では難しかった映像製品の開発効率と移植性の大幅な向上を実現するとともに、Webサービスと連動したGUIを容易に開発できるようになる。

4.1 WebKitの拡張

GUIのプラットフォームとしてWebKitを活用する場合、次のような、通常のブラウザとは異なる動作をすることが求められる。

- (1) 背景を透過し、映像と同時に表示できる
- (2) 複数のページを重ねて表示できる
- (3) 映像製品の機能を使用できる

当社は、独自に開発して実装したプラットフォーム依存コードや、WebCoreの機能を利用するアプリケーションコードにより、これらの要求を実現している。

特に、(3)の機能については、JavaScriptCoreの機能を活用することで、Webブラウザ上のJavaScriptから呼び出せる独自のAPIを追加している。例えば、チャンネル情報や録画コンテンツなどの情報をHTMLページから取得できるようにした。また、映像製品組み込みのGUIを実現するためには、映像製品から状態変更通知を受け取って表示する機能も欠かせないが、JavaScriptのDOMイベント^(注15)を用いて通知できるようにしている。

更に、JavaScriptのAPIをIDL (Interface Definition Language) の形式で記述することで、当社が開発したソースコードと接続する“グルーコード”を自動的に生成する機能も開発して搭載している。

4.2 組み込み機能の呼出し方式

今回の開発では、JavaScriptに独自のAPIを追加することにより映像製品の機能を呼び出す方式を採用したが、その他にも次のような方式がある。

- (1) NPAPI (Netscape[®] Plugin API) プラグインを使用する。

(注15) マウスクリックのようなユーザー操作イベントなどを規定したHTML関連規格の一つ。

- (2) HTTP (Hypertext Transfer Protocol) サーバを導入し、リクエストする。

- (3) HTML5のWebSocket機能を使用し、通信する。

これらの方式にはそれぞれ実行速度や、技術の普及度、アプリケーションの記述方法などに長所と短所があるので、製品戦略や実際の要求仕様に基づいて選択する必要がある。

5 あとがき

オープンソースのWebブラウザエンジンであるWebKitの概要と当社の取組みについて述べた。

当社は今後も、WebKitの軽量・高速化及び安定化に取り組むとともに、WebブラウザをGUIのプラットフォームとして活用するための研究開発を続ける。

文献

- (1) W3C. "HTML5 A vocabulary and associated APIs for HTML and XHTML". Working Draft 29 March 2012. <<http://www.w3.org/TR/2012/WD-html5-20120329/>>, (accessed 2012-07-20).
- (2) WebKit Project. "The WebKit Open Source Project". <<http://www.webkit.org/>>, (accessed 2012-07-20).



平野 裕 HIRANO Yutaka

デジタルプロダクツ&サービス社 プラットフォーム&ソリューション開発センター プラットフォーム・ソリューション開発第三部。組み込みGUIプラットフォームの開発に従事。Platform & Solution Development Center



深井 祐介 FUKAI Yusuke

デジタルプロダクツ&サービス社 プラットフォーム&ソリューション開発センター プラットフォーム・ソリューション開発第三部主務。組み込みGUIプラットフォームの開発に従事。Platform & Solution Development Center