

# Qtを組み込みで使う理由とQt 5

朝木 卓見

Qtは、デスクトップ以外にも組み込みLinuxをサポートしたアプリケーション(アプリ)開発フレームワークです。さまざまな製品でQtが使われ、日本では、シャープのLinux ZaurusやSONYのMylo、PanasonicのSkypeフォンなどが発売されています<sup>注1</sup>。以前のQtには商用版かGPL版しかなかったため、ライセンス料やソース・コードの公開などが課題でした。しかし、2009年にLGPLが追加されて以来、製品に無料で組み込みやすくなり注目が高まっています。

## ●Qtのメリット

それではQtを組み込みで使うメリットや、その場合の注意点は何か。まず、Qtのメリットを挙げてみます。

・**豊富な機能**：QtはよくGUIツール・キットとしてgtk+などと比較されます。確かに、QtにはGUIだけでも多くの便利な機能がありますが、対応範囲はそれだけではありません。マルチスレッドやコンテナなどの基本のクラス、OpenGL、ネットワーク、データベース、XML、マルチメディア、ECMAScript (JavaScript)、WebKitなどをQt APIでカバーしています(図1)。

もちろん、これらの組み合わせも簡単に行えます。また、アドオンとしてQt Mobilityというモバイル向けAPIも開発しています。これは、GPSや傾きセンサ、カメラ、Bluetoothなどに対応しています。機能によってはバックエンド用のプラグインの作成が必要ですが、MeeGo用のソースなどを参考にするといでしょう。

・**ソース・コードの公開**：Qtは、そのソース・コードが公開されています。ユーザはライセンスに従ってQtに変更を加えて使えますし、トラブル発生時の解析をしやすいです。

・**商用サポート**：Qt開発の中心を担っているNokia社自体は商用ライセンスの提供やサポートをしていますが、パートナー各社がサービスを展開しています。LGPLでライセンス料を抑えながら、コンサルティングやサポートを受けることも可能です。

・**移植性**：Qtは、組み込みLinux以外の主要なデスクトップ環境にも対応し、アプリのソースの多くを共有できます。QtのAPI自体は組み込みもデスクトップも共通なので、ソース・コード全体の共有が可能です。画面解像度やインターフェースなどが異なるため、UIはそれぞれに最適化した方がよいでしょう。また、コミュニティ・サポートですがVxWorksやQNXも対応しており、Digia社によるINTEGRARY RTOSのサポートや、SRAによるT-Kernelへの移植など、多種多様なプラットフォームで動作しています。

・**作りやすさ**：フットプリントの面でデメリットとされがちな

C++ですが、オブジェクト指向とGUIとの相性のよさは使ったことのある方にはおなじみでしょう。Qtは分かりやすいAPIと、GUI Designerを用いたWYSIWYGによるUI設計などによるアプリの作りやすさでは定評があります。最近では、後述するQML/Qt Quickの導入により、UIをより開発しやすくしています。また、デスクトップ環境やQVFBなどによるデバッグは、実デバイスでのテスト時のエラー軽減に役立ちます。

## ●Qtのデメリット

それでは、Qtを使うデメリットは何でしょうか。

・**フットプリントの増加**：QtはC++のランタイムが必要です。Qt自体のライブラリも小さなサイズではありません。それが問題となるケースもあるでしょう。けれども、Qtにはフットプリントを削減するためにqconfigと呼ばれる機能があります。これは、Qtライブラリから使用しない機能を削除することでフットプリントを削減する機能です。configureとビルド時に指定を行う必要がありますが、場合によってはかなりのサイズを削減できます。各ライブラリがどの程度のフットプリントを必要とするかの目安は、ドキュメント<sup>注2</sup>に示されています。

なお、ライセンス条件に問題がない場合、静的にリンクすることで不要な機能を実行ファイルに含まれないようにすることもできます。もっとも、LGPL版ではリンク用オブジェクト・ファイルの配布が必要なので、実質的には商用版の利用時に限定されます。

## ●Qtを組み込みで使う際の注意点

Qtを組み込みLinuxで使う場合、QWS (Qt Window System) やX11、DirectFBなどの環境で動かします。それぞれの環境でQtを動かすときの特徴を簡単に説明します。

・**QWS (Qt Window System)**：QWSは、組み込みLinuxでQtを使用する場合によく使われています。QWSでは、Linuxのフレーム・バッファをQtが直接ドライブするため、Linuxカーネルの標準の機能があればQtを動かします。X11などの中間層がないため、高速かつ省メモリでの動作が期待できます。

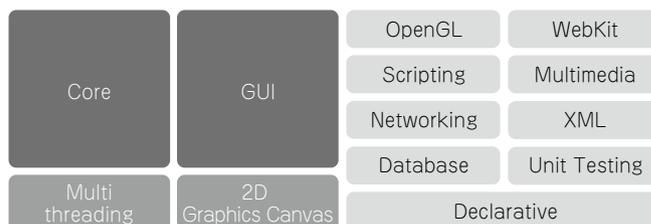


図1 Qtの豊富な機能

注1: <http://qt.nokia.com/qt-in-use-jp>

注2: <http://doc.qt.nokia.com/latest/fine-tuning-features.html>,  
<http://doc.qt.nokia.com/latest/requirements-embedded-linux.html>