

ネット公開を目的としたマルチウィンドウアプリ用フレームワーク

— GUI on the Net を実現する Ruby/TkORCA —

1 背景

現在，インターネットを通じての不特定多数へのサービス提供は Web を通じて行うのが一般的である．しかし，ローカルのウィンドウシステムの GUI と比べた場合には操作性や制約において大きな違いが存在する．現状では，ローカルの GUI と同等の操作性を提供しようとしても適切な手段がなく，ローカルでの利用とネットワークを介しての利用とで異なるインターフェースを作らざるを得ない．また，ローカルシステム用に非常に有用なライブラリが存在するにもかかわらず，Web サービスではそれが使えずに代用手法に頼らざるを得ないというケースもしばしば存在する．したがって，簡便さと安全性とを兼ね備え，ローカルのウィンドウシステムの GUI そのままにネットワークを介して利用可能なアプリケーションを作成するためのフレームワークが望まれる．

2 目的

本プロジェクトでは，オブジェクト指向スクリプト言語 Ruby の GUI 拡張のひとつである Ruby/Tk ベース (GUI の最表層が Ruby/Tk でありさえすればよい) でローカルのウィンドウシステム向けに書かれたマルチウィンドウの GUI アプリケーションを，ソースの変更はほとんど必要とせず，操作性はそのままに，セキュリティ確保のための監視・制御の下で，様々なクライアント環境の不特定多数向けに公開可能にすることが手軽にできるようにするためのフレームワークの開発を目的とした．

開発者はこのフレームワークを Ruby/TkORCA (Ruby/Tk on RFB Canvas) と呼ぶ．

Ruby/TkORCA により，GUI 作成技術がカバーする範囲をローカルからネットワークまで拡大して GUI アプリケーションのポータビリティを強化し，GUI を用いるネットワークサービスの提供を低コストで可能にすることを目指した．

3 開発の内容

Ruby/TkORCA を用いることで，サービス提供者はローカルのウィンドウシステム上と同等の GUI を簡単かつ安全に提供することができる．なお，Ruby/TkORCA のクライアント部分を Web ページ中に埋め込むこともできるため，Web との排他的利用ではなく融合させての活用が可能である．

Ruby/TkORCA が目指すコンセプトの一つは「どこでも GUI !! (GUI, Anywhere!!)」である．これは，クライアントに対する性能要求をできるだけ低くし，PC に限らず携帯電話や PDA クラスの機器からでも利用可能なものにするということを意味する．そのために Ruby/TkORCA では，サーバ・クライアント間の通信には RFB (Remote Frame Buffer) プロトコルを用いる．これは VNC (Virtual Network Computing) で用いられているプロトコルである．

Ruby/TkORCA の構造の概略図を図 1 に , Ruby/TkORCA を用いたサーバの構造の概略図を図 2 に示す .

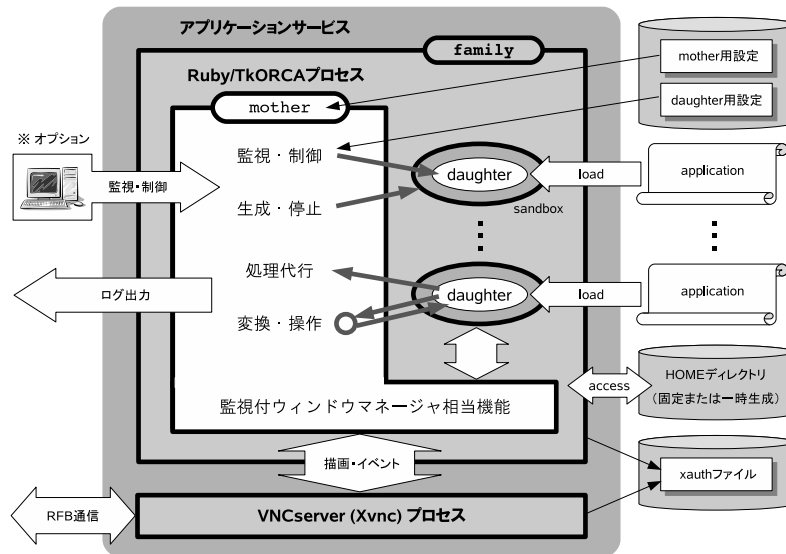


図 1 Ruby/TkORCA の概略構造図

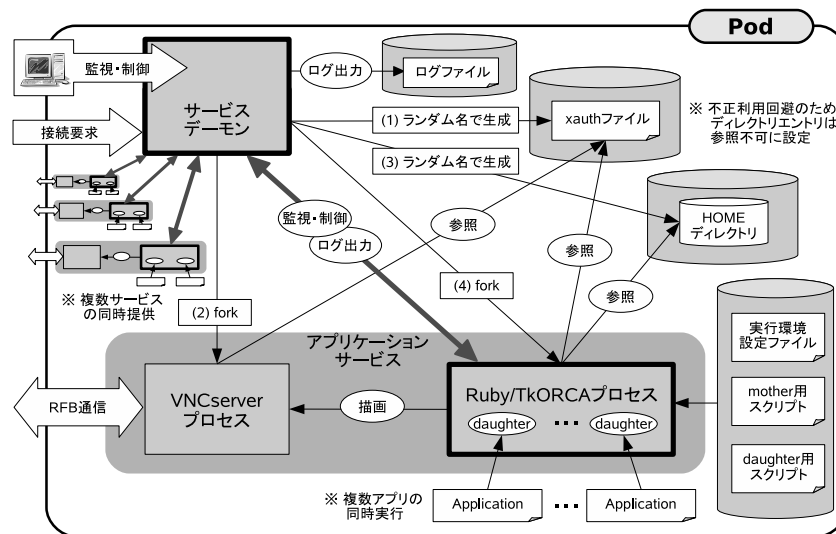


図 2 Ruby/TkORCA によるサーバの概略構造図

図のように , 現在の Ruby/TkORCA では一般の VNC サーバを活用してはいるが , これは RFB 通信ライブラリの代わりとして機能しているにすぎない .

Ruby/TkORCA は , 公開アプリケーションのためのウィンドウマネージャ相当機能および公開アプリケーションの監視や制御を司る mother 部と , 公開アプリケーションの実行環境である daughter (複数可) とから構成される . Ruby/Tk で書かれたスクリプトを daughter に読み込ませることで , mother による監視の下でそのスクリプトを公開アプリケーションとして外部から利用できるようになる .

公開するアプリケーションは , GUI の最表層が Ruby/Tk であれば良く , アプリケーションの規模には特に制約はない . 裏でどのような独自ライブラリを使ってい

てもかまわない．画面表示を行うものであっても，Tk のコンテナウィジェットに埋め込めるものであれば利用できる．また，shell を用いない仮想端末も用意しているため，コンソール入出力を用いるアプリケーションも安全に稼働できる．

ウィンドウマネージャ機能は，Ruby/Tk における canvas ウィジェットの上に構築されている．daughter で稼働中のアプリケーションのウィンドウをコンテナとなるフレームに埋め込んで管理することで，ユーザによるウィンドウ操作までも統一的に監視や制御をすることができる．なお，このウィンドウマネージャは仮想画面を持ち，実画面サイズよりも大きな範囲をスクロールさせて扱うことができる．

監視や制御は，実行状況に基づいて受動的に行うものだけでなく，管理者が能動的に介入して行うことも可能である．daughter で稼働中のアプリケーション内部に立ち入って操作することすら可能となっている．

Ruby/TkORCA はそのほとんどが Ruby のみで構築されており，サーバの動作環境として必要とされるのは次のものである．

1. Xvnc

inetd 対応 (標準入出力を RFB 通信に用いることができる) のものであること．Xvnc(またはその同等品) は何種類か存在するが，個別の画面を持つことができ，Xauthority 認証機構が利用可能なものであれば，いずれでも構わない．通信量圧縮の点からは Tight VNC を推奨する．

2. Tcl/Tk

上記の Xvnc に表示できるものであること．現時点の Tcl/Tk の開発状況から見て，Tcl/Tk 8.4 系を推奨する．

3. Ruby

1.8.5 以降，または 1.8.5 以降の Ruby に付属の Ruby/Tk をバックポートしたもの．Ruby/Tk の構成要素である tcltklib が上記の Tcl/Tk のライブラリを用いるように適切にコンパイルされていること．

その他，公開するアプリケーションに応じて，必要とされるライブラリ類が適切に導入されていればよい．特筆しておくべき例としては，一般的な Xvnc を用いて OpenGL を使うアプリケーションを提供する場合，3D サポートの関係上，OpenGL 互換ライブラリの Mesa の導入が必要となる可能性がある．

サービスの提供を受けるクライアント側の動作環境は，「いずれかの VNC ビューアが稼働すること」ということの他には条件はない．

4 従来の技術との相違

ネットワークを介した GUI 利用の他の手法は，信頼できるサーバ・クライアント間しか想定しておらず，不特定多数へのサービスでの利用は極めて難しい．

現状でネットワークを介したリッチサービス提供と言えば Ajax ということにな

と思うが，Ruby/TkORCA では Ajax の課題である，サーバ，クライアント双方でのプログラム開発を要することによるコスト増大という問題を避けられる．また，「ブラウザ上での動作」という足かせがないため，有用なライブラリを自由に用いてアプリケーションを構築することができる．クライアントのアクションに依存した受動的動作だけでなく，サーバ側からの能動的で即時の働きかけも可能である．

5 期待される効果

重いブラウザを必要としないため，能力の低いクライアントに対して容易にリッチサービスを提供することができるようになる．

クライアントに新しいアプリケーションやライブラリのインストールを要求することなく，特殊な機能を提供できる．これは，様々なクライアント環境を想定してのテストの必要性をなくすため，高度な新規サービスの提供を低コストで迅速にできることを意味する．

高速化するネットワークを動画コンテンツ以外で効果的に活用するための一つの形態となる．これにより，ビジネス用途などでの利用で動画試聴に興味がなく，高速ネットワークにさほどの必要性を感じていなかった利用者において，新たな高速ネットワーク需要を創出することができる．

業務等での専用アプリケーションで利用した場合，最新 OS を動かすには能力不足となった旧式機をクライアントとして有効活用できる．また，ディスクレスの小型軽量端末で十分であるため，それを店員に持たせるようにすれば，顧客との相談や確認もその場で素早く行うことができ，顧客の満足度の向上にも繋がる．

6 普及 (または活用) の見通し

Ruby/TkORCA は，個人利用から規模の大きなサービス提供までの様々なレベルで活用することができる．例えば，ちょっとしたパーソナルツールを遠隔利用や研究などで部外秘のライブラリを使ったデモシステムの公開，シンクライアントを用いるアプリケーションサーバなどである．

ビジネス用途の場合，クライアント側に情報を送らないために端末盗難などによる情報漏洩のリスクが低い．また，利用者によるマウス移動やウィンドウ操作までもモニタリングできる特性から，利用者の注目点や嗜好の分析にも活用でき，新たなビジネスに繋げることができる可能性もあるだろう．

7 開発者名 (所属)

永井 秀利 (九州工業大学情報工学部知能情報工学科)

(参考)

開発物の公式ページは未完成だが，次の場所で開発物の β 版の入手が可能である．

<http://qwik.jp/ruby-tk/RubyTkORCA.html>