

ソフトウェア機能テストを対象とした ファームウェアシミュレーターの開発

Development of a Firmware MFP Simulator for the Evaluation of MFP Software

松島 紀之* 山下 功* 西元 真司*
Noriyuki MATSUSHIMA Isao YAMASHITA Shinji NISHIMOTO

要旨

コニカミノルタソフトウェア研究所の主力事業であるコニカミノルタ製品のソフトウェア評価は、評価効率化策として中国での評価拡大を実施中である。国内と中国に拠点を分散して評価する際の重要施策として、評価機材レスの評価を進めている。この評価機材レスに向けて、MFPのファームウェアシミュレーターを開発した。

プロダクションプリントMFP用外付けコントローラ評価は実機にて実施していたが、実機の中国への輸送の制約やメンテナンス、構成変更の工数がかさみ、評価時間を圧迫していた。その対応策として、既存のWindowsベースのシミュレーターにGUIとVIF (Video Interface) ボードを追加した評価用のシミュレーターを開発し、評価機材レスを実現した。

また、オフィスMFPファームウェアには、実機OSと異なるWindows上で動作するシミュレーターがあるが、OSの違いにより搭載できないモジュールがあり、評価範囲に限界があったため、実機と同じLinux OS上で動作するシミュレーターを、汎用PCを用いて開発し、評価範囲を大幅に拡大した。

これらの開発は、短期開発と再利用性を重視し、シミュレーターと評価自動化システムを分離して開発した。結果、開発した評価自動化システムはすべてのシミュレーターで流用可能となり、カスタマイズを容易にして、評価現場の要望を取り入れ、生産性を向上させた。

Abstract

The evaluation of software for use in Konica Minolta MFPs, which is a core business of the Konica Minolta Software Laboratory, is being expanded in China to improve the efficiency of software evaluation. However, transporting MFPs to dispersed offices in China (and in Japan, as well) simply for software evaluation is impractical and prohibitive. In response, we decided to create a firmware MFP simulator that would do away with the need for an actual MFP in software evaluation.

The evaluation of an external controller of a production printing MFP had been carried out in China using an actual MFP. But we learned that, besides the problem of transporting MFPs to China, the man-hours of maintenance and configuration change also increased, which limits evaluation time. In response, we developed an MFP simulator for software evaluation in which GUI and VIF (video interface) boards were added to an existing Windows-based simulator. No longer was an actual MFP needed.

There already existed an office MFP simulator running on Windows, but the Windows OS differed from the OS of the MFP firmware, and there was a module which could not be installed due to this difference of OSs. This meant limiting the scope of possible evaluations. Therefore, we have developed a simulator, using a general-purpose PC, which runs on the same Linux OS as actual MFPs. This has greatly expanded the scope of evaluations.

During development, we strove for rapid progress and for reusability, and we separately developed a simulator and an automated evaluation system. The newly developed automated evaluation system can be utilized with simulations of all MFP systems, which makes customization easy and evaluation productivity improved.

執筆者



松島紀之



山下 功



西元真司

*コニカミノルタソフトウェア研究所(株) 第1開発部

1 はじめに

コニカミノルタソフトウェア研究所の主力事業であるコニカミノルタ製品のソフトウェア評価は、評価効率化策として中国での評価拡大を実施中である。拠点を分散して評価する際の重要施策として、評価機材レスの評価を進めている。この評価機材レスに向けて、MFPのファームウェアシミュレーターを製品群ごとに異なるアーキテクチャで開発した。また、シミュレーターと連動する評価自動化システムを開発し、評価の効率化を行った。

以下に、シミュレーターと評価自動化システムの構成と特徴について述べる。

2 概要

評価用のシミュレーターは中国への輸出に問題が発生しない構成とし、MFPの挙動を再現する部分は、実機との等価性を保証するために、当該ファームウェアの開発元によって開発されたシミュレーターや、ファームウェアそのものを実装したシミュレーターを利用した。

評価対象は、プロダクションプリントMFPの外付けコントローラーと、オフィスMFPのファームウェアとし、評価部分を明確にして、システム設計を行った。

また、開発工数の効率化のために、以下の点に留意しながら開発を行った。

- ①エンジンのシミュレーションと評価自動化システムの連動を実現するための拡張性の確保
- ②評価効率化に向けた操作性の改善のため、評価者の要望を取り入れ、ユーザビリティ向上を容易にするカスタマイズ性の確保
- ③評価自動システムを多機種に対応させ、最小限の開発費で最大限効率化できるように、機能モジュール化により、他のシミュレーターへの転用性を確保

3 プロダクションプリントMFPシミュレーター

3.1 開発の背景と目的

プロダクションプリントMFPの評価において大きな課題となっていたのは、評価機材の調達である。試作段階の機材は数の確保が難しいだけでなく、機材の輸送の困難さや、メンテナンス・構成変更に多くの時間を必要とすることが、評価効率の低下の原因となっていた。

プロダクションプリントMFPでは、Windowsで動作する開発用のシミュレーターが存在しており、論理動作の確認に使用されていた。このシミュレーターを利用して、外付けコントローラーの評価に対応できるシミュレーターを開発し、機材の使用台数を削減した。

3.2 開発によって達成すべき目標の設定

- ①評価に使用する機材を50%以上削減する事
- ②評価項目の70%以上をカバーし、等価性を向上させる事

- ③Windows上で動作し、外付けコントローラーから見たエンジン動作を、印字速度を含めてすべてカバーし、実機の代替が可能な事

3.3 システム構成

プロダクションプリントMFPシミュレーターの評価対象は、外付けコントローラーである。Fig. 1に構成図を示す。

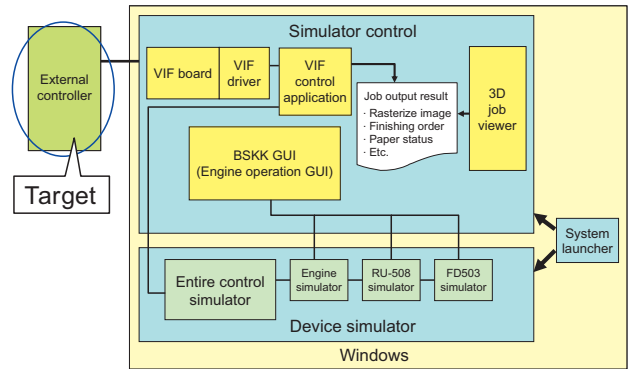


Fig. 1 Production printer MFP simulator.

プロダクションプリントMFPシミュレーターはMFP本体及びフィニッシャーをシミュレーターに置き換え、それらを1台のWindowsPC上で動作させる構成とした。以下が新規に開発したモジュールである。

- 汎用PC用VIFボード (VIF Board)
- VIFボード制御用デバイスドライバー (VIF Driver)
- VIFボード制御用アプリケーション (VIF Control Application)
- シミュレーター操作用GUIアプリケーション (BSKK GUI)
- プリントJOB表示用アプリケーション (3D Job Viewer)
- システム起動用アプリケーション (System Launcher)

外付けコントローラーの出力画像を600Dpi=>150Dpiに間引きするFPGAロジックをVIFボードに搭載し、その画像をPCI-Expressバス(4レーン)経由にてスキャッター&ギャザーDMAを使用して、PCのメモリへ高速転送することで、外付けコントローラーの画像出力速度に対応した。

これにより、出力画像(A4フルカラー600Dpi)をPCのメモリへ転送する時間は、1ページあたり約600msec(毎分97ページ)となり、150Dpiへの圧縮を行った場合には、PCへの転送時間が1ページあたり約40msec(毎分1500ページ)で、現在市販されている製品の最高出力速度(毎分85ページ)を超える出力速度に対応できた。

3.4 開発上の工夫

評価に適用するシミュレーターシステムでは、プロダクションプリントMFPの実機と本シミュレーターとの等価性が重要である。

また、評価の効率向上のためには、ユーザビリティと評価自動化への対応も必要となる。本シミュレーターシステムは、これらの点を考慮して開発した。

3.4.1 実機ファームウェアロジックとの等価性保証

本シミュレーターシステムにおいては、プロダクションプリントMFP本体のシミュレーター及びフィニッシャーのシミュレーターについて、Windowsで動作する実機ファームウェアを搭載したシミュレーター(以下、デバイスシミュレーター)を使用した。

デバイスシミュレーターは実機ファームウェアの開発で論理動作検証に使用されているシミュレーターで、実機と等価のファームウェアロジックを搭載している。このため、デバイスシミュレーターは実機ファームウェアロジックとの等価性が保証されている。

3.4.2 ハードウェアI/Fの等価性保証

外付けコントローラーとの接続でプロダクションプリントMFPとの等価性を保証するため、接続用ケーブルについても実機と同じケーブルを使用可能とした。

コマンドの通信に使用するシリアルI/Oチップやボーレートジェネレーター用オシレーターについても実機と同じ部品を使用した。これらの対応をVIFボードで行った。

3.4.3 フィニッシャーの構成組み替えを容易に実現

プロダクションプリントMFPの実機を用いた評価においては、フィニッシャーの構成変更が必要となり、多大な工数と労力を費やしている。

本シミュレーターシステムにおいては、システム起動用アプリケーション(System Launcher)で、フィニッシャー構成を選択し、様々なフィニッシャー構成で起動することができる。

3.4.4 多様なフィニッシャーの出力結果を可視化

パンチ/ステープル/綴じ/折りのフィニッシャーの処理結果をプリンターコントローラからの出力画像に合成&表示して、各プリントJOBについてのフィニッシャーの出力結果をプリントJOB表示用アプリケーション(3D Job Viewer)で確認可能とした。

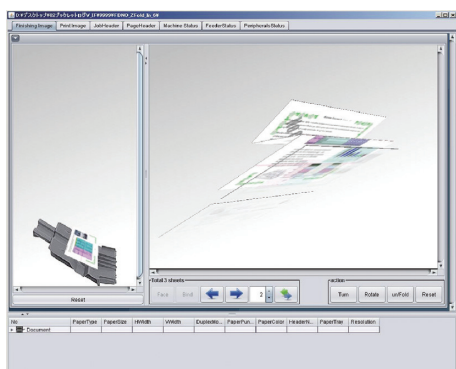


Fig.2 Z-fold display view for confirmation of output result.

また、このプリントJOB表示用アプリケーションでは、フィニッシャーの出力結果だけでなく、出力先トレイの確認や、出力先トレイへ出力された用紙の向き、折りの状態、中綴じ、パーフェクトバインド製本などの出力結果の確認が行えるようにした。

例としてZ折りの出力結果確認画面をFig.2に示す。

3.4.5 評価自動化への展開

デバイスシミュレーターはGUIアプリケーションからの操作を可能にしてユーザビリティを向上させた。

この操作には

- ・アラーム検出センサーのステータス変更
 - ・各用紙トレイにおける用紙の交換&補充&撤去
- 等があり、GUIアプリケーション以外からの操作を容易にするためのSDK(Software Development Kit)も併せて開発した。

このSDKを使用したアプリケーションを作成する事で、用紙やトナーの補充を連続して行いながら、自動的に評価を行う事が可能となった。

3.4.6 実機イメージを使用したGUIデザイン

GUIアプリケーションは、実機イメージをそのままグラフィック表示している。(Fig.3参照)

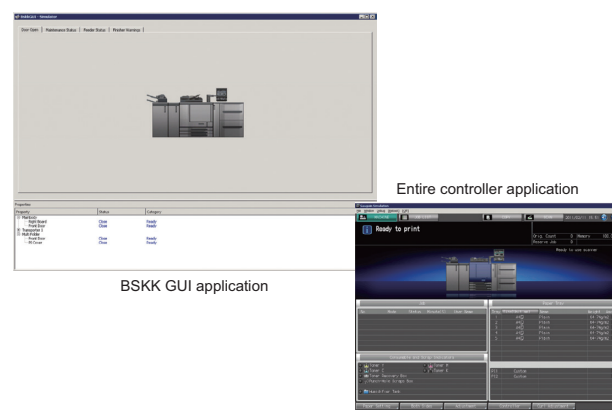


Fig.3 GUI of production print MFP simulator.

これにより、プロダクションプリントMFPの実機を使用した評価の経験者であれば、特別な事前教育を必要とせずにGUIが操作できる。

さらに、実機を使用した評価を行う前に、本シミュレーターシステムを利用して最新のプロダクションプリントMFPの実機の操作を学ぶ事が可能になった。

3.5 目標の達成状況

- ①評価に使用する機材の削減率は93%であった。
- ②評価項目のカバー率は73%であった。
- ③作成したシミュレーターはWindows上で動作し、外付けコントローラーからは、印字速度を含め、エンジンと等価に動作し、評価への投入が可能なレベルとなった。

4 オフィスMFPシミュレーター

4.1 開発の背景と目的

オフィスMFPの評価で最も課題となっていたのは、機材の調達である。試作機は非常に高価で数量の確保が難しく、一つの機材を複数の開発者と評価担当者が共有して作業にあたる事もあり、効率の悪さが目立っていた。

オフィスMFPではWindowsで動作するシミュレーターが存在していたが、実際のオフィスMFPで使われるOS(リアルタイムLinux)との違いが大きく、各種動作の再現性が悪いため評価に投入できるレベルではなかった。

本シミュレーターの開発は、bizhub C754シリーズの評価において評価に耐えうる等価性を満たし、機材の使用台数を削減する事を目的とした。

4.2 開発によって達成すべき目標の設定

- ①評価に使用する機材を50%以上削減する事
- ②ファームウェアのリリースから5日以内にシミュレーターをリリースできる事
- ③汎用のx86 PCで使用出来る事
- ④ネットワーク評価(PSWCとOpenAPI)に適用できる事

4.3 システム構成

4.3.1 オフィスMFPファームウェアのシミュレーター

オフィスMFPのファームウェアでは、x86のリアルタイムLinuxで動作するシミュレーターとして、以前より開発されていた bizhub C652シリーズのシミュレーターがあり、この研究成果を次世代機であるC754シリーズへと取り込んだ。これによりx86 Linux PCでオフィスMFPファームウェアをビルド・実行する事が可能になった。

4.3.2 周辺機器のシミュレーター

パネル、スキャナー、プリンターなどの周辺機器のシミュレーターはWindows上で動作するものが既に開発されていたが、Linux上で動作出来るものはない。

そのため、周辺機器シミュレーターはWindows用のものをそのまま使用し、Linuxで動作するオフィスMFPシミュレーターとはネットワーク通信を経由して接続するように構成した。Fig. 4に構成図を示す。

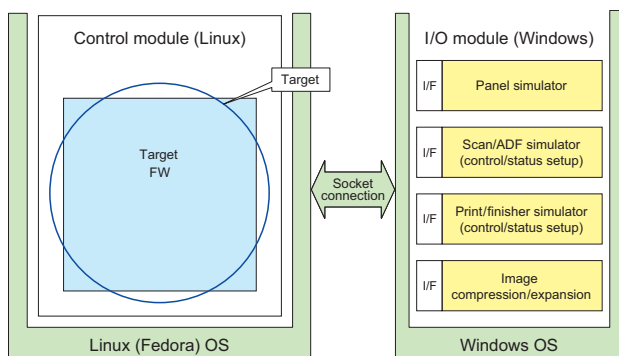


Fig. 4 Office MFP simulator.

4.3.3 仮想マシンの併用

オフィスMFPファームウェアと周辺機器のシミュレーターは別々のOSで動作するため複数のマシンが必要になる。最小限の環境で動作出来るようにするために、オフィスMFPファームウェアのシミュレーターを仮想マシンのゲストOSとして動作させ、周辺機器シミュレーターはホストOSであるWindowsで動作させるようにした。

4.4 開発上の工夫

4.4.1 ライセンス料コストの削減

実際の機材で使用されているLinuxシステムにはライセンス契約により一部無償では使用出来ない部分が含まれているため、多数の開発者、評価担当者のためのライセンス料を節約する必要がある。

そのために、ベースとなるOSには無償で利用出来るFedora Linuxを採用し、Linuxカーネルは実際の機材で使用しているものと同じリアルタイムLinuxに入れ替えた。またライブラリやツールチェーン(binutils, gcc, glibc, etc..)等も互換性のためバージョンを合わせたものに入れ替えている。

4.4.2 仮想マシン利用による利便性向上

仮想マシンには無償で利用出来るVirtualBoxを採用した。VirtualBoxにはハードディスクのスナップショット機能があるため、ファームウェアのバージョン別に保存したり、オフィスMFPのある時点の設定を保存出来るため、開発や評価での利便性が向上した。

また、仮想マシンを利用することにより、Linuxで使用するデバイスドライバの種類が減り、実際のPCで使った場合と比べてLinuxカーネルの検証、保守、運用にかかる工数を削減する事が出来た。

4.4.3 RPMを使ったバイナリ管理で保守性の向上

従来はファイルのコピーといった単純な方法で更新を行っていたが各ファイルの不整合が起りやすく、特にLinuxでの操作は評価担当者には扱いが難しかった。

Fedora LinuxはパッケージマネージャーRPMでソフトウェア構成を管理しているため、不用意なファイルのコピーは既存システムを破壊してしまう恐れもあった。

そこで、オフィスMFPファームウェア、Linuxカーネル、ライブラリ、ツールチェーンなどのバイナリファイルもRPMのパッケージファイルとして作成し、RPMファイルのみで配布するようにした。これにより確実なインストール、アンインストールが行えるようになり、バージョン不一致といったミスが起りにくくなった。

4.4.4 シミュレーターの操作簡略化

シミュレーターの起動、停止、バージョンアップなどには複雑なコマンド操作が必要であり、かつWindowsとLinuxで同時に操作する必要があるため、評価担当者には扱いが難しかった。

これを改善するため、Windows側からバッチファイルをクリックするとWindowsとLinuxへのコマンド操作を自動的に行うスクリプトファイルのセットを作成した。

WindowsからLinuxへのリモートコマンド操作にはCygwinのSSHなどパッケージの一部を利用し、シェルスクリプトで制御を行えるようにしている。

• setup.bat

接続するシミュレーターのIPアドレス設定を行う。同時にSSHの公開鍵をLinuxへ転送する。この設定により以降Linuxに接続する際にパスワードの入力が不要になる。

• mfp-run.bat

オフィスMFPシミュレーターを起動する。Windowsでパネル、周辺機器シミュレーターを起動し、その後Linuxでシミュレーター起動のコマンドを実行する。パネルシミュレーターのウィンドウを閉じると同時に周辺機器シミュレーターを終了し、Linuxでもシミュレーター停止コマンドを実行する。

• update.bat

オフィスMFPシミュレーターのバージョンアップを自動的に行う。カレントディレクトリにある更新用のRPMファイルをLinuxに転送し、その後Linux側でRPMを更新するコマンドを実行する。

4.4.5 コンパイルの効率化

オフィスMFPのファームウェアのコンパイルには8時間前後の時間を要していたため、リリースの遅れやデバッグの障害となっていた。この問題を解決するためccacheとdistccを採用した。

ccacheはコンパイラキャッシュ機能を提供するフリーソフトウェアである。コンパイル対象のファイルに変更がない場合、キャッシュされた前回のコンパイル結果を返すことでコンパイル時間を短縮する。

distccは複数のPCにて分散コンパイルを行うフリーソフトウェアである。指定したPCを使って並列にコンパイルを行うため台数を増やした分だけコンパイル時間が短縮される。

ccacheとdistccを組み合わせることでキャッシュされていないファイルのみを分散コンパイルの対象と出来るため、リビルドにおいてもコンパイル時間を短縮する事が出来る。これらの対策によりコンパイル時間は従来の8時間から100分前後まで短縮した。

さらに分散コンパイルのためのdistccサーバーを簡単に設置出来るようにするため、予めdistccをインストール、設定を行った仮想マシンのイメージファイルを作成した。これにより空いているPCにイメージファイルをコピーし起動するだけでdistccサーバーとして機能させ、分散コンパイル環境が容易に構築できるようになった。

4.4.6 ソースコード管理の効率化

従来のバージョン管理システム（Subversion, Clear Case）での各種操作にはサーバーへの通信が必要である

ため、オフィスMFPファームウェアの巨大なソースツリー（約4GiB前後）に対してマージ操作を行うのは非常に時間がかかり、困難であった。そこで各種操作にサーバーへの通信を必要としないバージョン管理システムgitを採用した。

gitとはLinuxカーネルプロジェクトで開発されたフリーソフトウェアである。分散型のバージョン管理システムで、ローカル環境のみでのコミット、ブランチをサポートし、分散された複数のリポジトリと任意に同期出来るのが特徴である。

gitのネットワークを介しない高速な操作により、従来1時間前後かかっていたマージ操作を10分前後にまで短縮出来るようになった。

4.5 目標の達成状況

- ①評価に使用する機材を55%以上削減
- ②ファームウェアのリリースに対して3日以内に同じレベルのシミュレーターをリリース
- ③Core2シリーズ以降の汎用のx86 PCで使用出来る事を確認
- ④PSWCおよびOpenAPIの評価業務に投入し、効率化

5 評価自動化システム

5.1 開発の目的

評価用のシミュレーターは操作をソフトウェアで制御できるI/Fを搭載した。これを利用して評価の自動化を行い、評価工数の削減を行う。

ここでは、プロダクションプリントMFPのMIB評価の自動化による工数の削減を説明する。

5.2 開発によって達成すべき目標の設定

プロダクションプリントMFPのMIB評価自動化について、以下の目標を設定する。

- ①評価時間の削減率15%以上。
- ②評価カバー率70%以上

5.3 システム構成

プロダクションプリントMFPシミュレーターに、連動して動作するMIBの評価自動システムを追加した。Fig.5に構成図を示す。

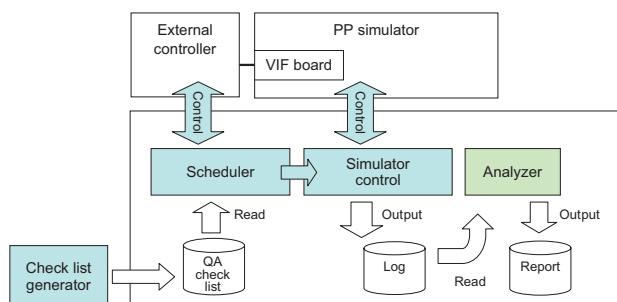


Fig.5 Automated evaluation system and MFP simulator.

5.4 開発上の工夫

MIBの評価システムは、シミュレーター操作のI/FのSDK化と標準化により、オフィスMFPのシミュレーターでも利用可能にした。

また、チェックリストを標準化し、チェックリストジェネレータを作成して、機種対応はジェネレータで行うことで、チェックリストの読み込みを自動化して、評価自動化システムの流用性を高めた。

5.5 目標の達成状況

①評価時間の削減率17.3%

②評価カバー率100%

③評価項目数 実機：5914項目（手動のみの場合）

→本システム：87049項目（自動の場合）

従来、実機による評価では、フィニッシャーの組み換えに時間が多くかかったため、評価時間の制約から全評価項目の6%程度（5914項目）に絞って評価を行っていた。自動化システムとシミュレーターの組み合わせでは、終夜運転で全項目（87049項目）を毎回確認でき、リグレッションの検出など、これまでにできなかった評価が実現できた。

6 まとめ

プロダクションプリントMFPおよびオフィスMFPのシミュレーターの基本モジュールが整備、部品化できた。

特にFPGAを使ったVIFボードにより、従来難しかったハードウェアのタイミングに関する評価が可能になり、活用範囲が大きく広がった。

また、WindowsとLinuxを混合した形でシステム構築するためのノウハウが蓄積できたので、新たな評価対象への対応が容易になった。

現在、プロダクションプリントMFPで、Linux上で動作する自社製コントローラーを対象とした評価システムを開発しているが、これまでに開発した外付けコントローラーの評価システムに、オフィスMFPシミュレーターで培ったLinux上のシミュレーターのノウハウを導入して開発を進めている。

今後も、対応範囲を拡大して、評価コストの削減に貢献していきたい。

●使用したフリーソフトウェア

Fedora Linux - <http://fedoraproject.org/>

VirtualBox - <https://www.virtualbox.org/>

Cygwin - <http://www.cygwin.com/>

ccache - <http://ccache.samba.org/>

distcc - <http://code.google.com/p/distcc/>

Git - <http://git-scm.com/>