

次世代ATMソフトウェア

森田 満二

近年のWeb技術の発展は、複数システム間の連携を容易にし、個々のシステム単体では実現困難であった付加価値の高いサービスの提供を可能にした。また、利用者は携帯電話に代表されるさまざまなサービスを受ける機会を身近に持つことが可能となっており、今後、ユビキタスサービスとしてさまざまなサービスが実現されてくると予想されている。金融システムにおいても、さまざまなシステムと連携し、ユビキタスサービスの提供を可能とするため、Web系技術をベースとした基盤整備が進められている。ATMとしては、Web系技術の特長を活かすためのATM（以下WebATM）が各社から発表されている。

主なWebATMの特長は、以下の3つである。

特長1：他システム連携

Web系技術の採用により、他システムとの連携が容易になる。

特長2：プログラムのサーバ配置

プログラムのサーバ配置により、ATM個々にアプリケーションを配置する必要がなくなる。

特長3：データのサーバ集約

ATMの運用に必要なジャーナル情報やスケジュール情報などをセンターで一括管理可能になる。

しかし、Web系技術とATM技術の融合には課題が多く、本格的にユビキタスサービスを提供するに至るWebATMは開発されていない。そこで我々は、これら3つの特長を維持しつつWeb系技術とATM技術を融合させ、本格的にユビキタスサービスを展開可能とする次世代ATMフレームワーク（以下次世代ATM-FW）を開発した。次世代ATMソフトウェアは、この次世代ATM-FWを利用し開発されている。ここでは、次世代ATMソフトウェアの技術的特長を、次世代ATM-FW中心に紹介する。

Web系技術のATM適用における課題

一般的なWebシステムの処理イメージを図1に示す。

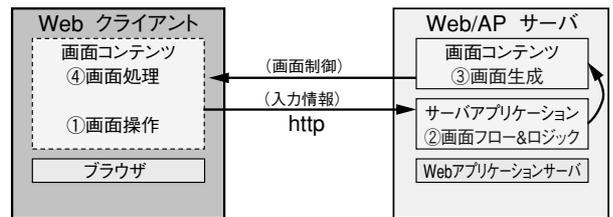


図1 一般的なWebシステムの処理

一般的なWebシステムは、図1の①から④までの画面処理（ブラウザ画面上の入出力制御とサーバ側でのロジックおよび次画面の生成）を繰り返すことによりアプリケーションが実行される。一般にWebは、さまざまなWebシステムに接続（特長1）し、都度アプリケーションをサーバからダウンロード（特長2）することで実行する。しかし、この仕組みは、悪意あるシステムに接続した場合、悪意あるプログラムがクライアントで実行され個人情報流出等のセキュリティに関する問題を多く生んできた。このため、Webクライアントとしては、画面処理のみにブラウザの設定で制限されており、クライアント内のローカルデバイスを制御できない様にしている。このWeb処理の流れにATMとしての処理を実装した場合、図2に示す流れとなる。

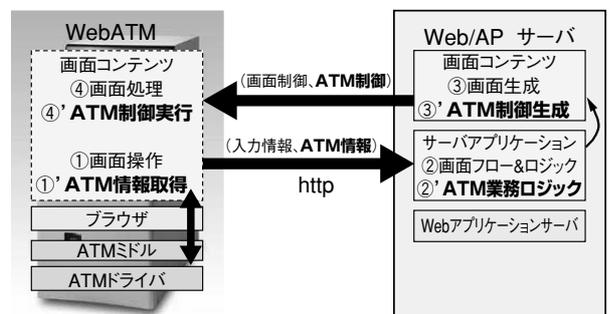


図2 一般的なWebATMの処理

このように単純にWebの処理に合わせると、①から④の画面処理の他に①'から④'までのATM制御（ATM上のカード・レシート・ジャーナル・硬貨・紙幣・通帳等のさまざまなデバイス制御およびサーバ連携によるホスト通信やDB処理）が必要となる。

ここで、一般的なWebATMの課題をまとめる。

（課題1）セキュリティの設定レベルの低下

図2に示す構成のWebATMでは、ローカルデバイスの制御を必要とするATM制御をブラウザ上から行う必要がある。図2に示す構成のWebATMとしては、ブラウザのセキュリティの制限を解除するために、セキュリティの設定レベルを下げる必要が出てくる。ATMには、紙幣ユニットの制御など現金を扱うローカルデバイスが存在する。ブラウザからローカルデバイスの制御を必要とするようなプログラム構造は、将来のさまざまなユビキタスサービスの展開の際にATMを多くの脅威にさらすこととなる。このためWebATMとしては、ブラウザの設定を下げずにローカルデバイスを制御する仕組みが必要となる。

（課題2）通信量の増大

既存のATMは、ATM内で多くの処理を実行し、ホストとの通信時にのみ外部との通信が発生する。図2に示す構成のATMでは、ほぼ画面にATMとサーバ間の通信が行われ、通信量の増大を招く。一般的にWebシステムは、画面操作の度にサーバ通信を行うと非効率であるため、キャッシュを利用する仕組みやより多くの処理をクライアント側で実行するためのクライアント制御技術が発展（JavaScriptからAjaxへの発展等）してきた。一方、既存のATMのネットワークは、従来からの低速なネットワークによりセンターと接続されている場合が多い。Web系技術を採用したATMを展開するために、ネットワークインフラを高速なものに整備し直すことは、現実的ではなく市場に浸透しにくい。そこでATMにWeb系技術を採用する際は、クライアント制御技術を利用し、通信量を最小限に抑える工夫が必要となる。

（課題3）画面コンテンツとデバイス制御が強く依存

ATMにユビキタスサービスを展開するには、ATMの画面上に自由にユビキタスサービスの画面コンテンツを表示できることが望ましい。ATM制御との連携は、カードの出し入れやレシートの排出といった機能をユビキタスサービスの画面コンテンツ側が制御しやすい単純な最小限の機能であることが望まれる。しかし、図2に示す構成において、ATMの画面処理とATM制御は、同時に実行さ

れ強い依存関係を持つ。この構成では、ユビキタスサービスの画面コンテンツの開発者にATM制御を含め開発させることとなり、特長1に示すように容易にユビキタスサービスを展開する事が困難になる。ATMをユビキタスサービスに対応するには、画面処理とATM制御を分離する必要がある。

次世代ATMソフトウェアの概要

次世代ATMソフトウェアの構成図を図3に示す。

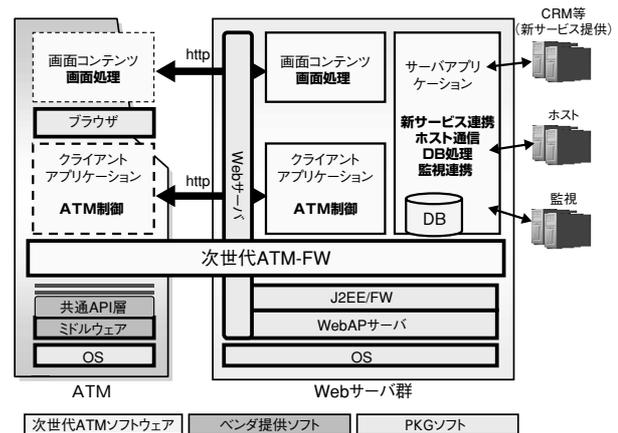


図3 次世代ATMソフトウェア構成図

次世代ATMソフトウェアは、画面コンテンツとクライアントアプリケーションおよびサーバアプリケーションに分類される。画面コンテンツとクライアントアプリケーションは、クライアント制御技術で実装されている。また、画面コンテンツとクライアントアプリケーションは、サーバ側にプログラムが配置され、プログラム更新時にサーバ側からATM側にダウンロードされキャッシュされる。画面コンテンツは、画面処理を行い、クライアントアプリケーションは、ATM制御を行う。また、サーバアプリケーションは、ホスト通信やDB処理の他、新サービスへの連携や監視システムへの連携を行う。次世代ATM-FWは、これらの画面コンテンツ、クライアントアプリケーション、サーバアプリケーションに対し、汎用的なプログラム構造や機能を提供している。一般にソフトウェア開発では、生産性や品質の向上を目的に、対象とするアプリケーションにおいて汎用的に利用可能なプログラム構造や機能を体系化したフレームワーク（Struts等）が利用されている。次世代ATMソフトウェアとしても、ATMシステムとして必要となる取り引きのフローや各種デバイスの制御に対し汎用的なプログラム構造や機能を定義し次世代ATM-FWとして体系化した。

次世代ATM-FWの機能

次世代ATM-FWは、次世代ATMソフトウェアに対して汎用的なプログラム構造や機能を提供するだけでなく、1章に示すWebATMの課題解決を行っている。

次世代ATM-FW適用時の処理を図4に示す。

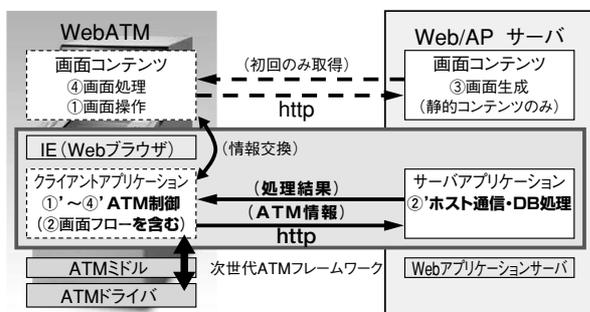


図4 次世代ATM-FW適用時の処理

主な次世代ATM-FWの機能は、以下の5つに分類される。

(1) 情報交換機能

課題1の対策としては、ブラウザ上でのATM制御を避け、クライアントアプリケーション上でATM制御を行うようにする。画面処理とATM制御は、情報交換機能で連携する。情報交換機能は、ブラウザのセキュリティ上の制限を受けずにhttp通信を活用することで、セキュリティの問題を回避している。画面コンテンツとしては、一般的なクライアント制御技術のJavaScriptを記述するのみで開発可能となり、ローカルデバイスを制御するためにActiveX等のセキュリティ設定を下げる必要のある技術の利用から開放される。

(2) クライアントアプリケーションの実行機能

課題2の対策として、クライアントアプリケーションの実行機能は、サーバ側との間の通信を最小限にしつつATM制御を可能とする仕組みをクライアントアプリケーションに提供する。クライアントアプリケーションの実行機能は、一般のWebシステムのように画面表示ごとにサーバアクセスすることなく、クライアントアプリケーションをプログラム更新時にまとめてキャッシュとしてATM内に取り込みサーバアクセスなしに実行させる機能を提供する。

クライアントアプリケーションは、図4に示す通り、クライアントアプリケーションの実行機能により①'～④'のATM制御および、②画面フローを実現する。クライ

アントアプリケーションとサーバアプリケーションとの通信は、特長3に示したデータのサーバ集約の特長を維持するのに必要な処理（ホスト通信やDB処理）のみとなる。

(3) ブラウザ制御機能

ブラウザ制御機能は、課題2および3の対策のために、クライアントアプリケーションからの、画面コンテンツに対する画面表示依頼の制御を可能にしている。クライアントアプリケーションは、画面フローも含めて実装しているが、画面コンテンツの表示そのものは、次世代ATM-FWのブラウザ制御機能を通して行う。画面コンテンツは、ブラウザ制御機能で依頼された画面をサーバから取得することで画面表示を行う。この時（1）画面コンテンツとクライアントアプリケーションの情報交換機能により、画面上に表示する金額等の情報は、ATMでローカルにクライアントアプリケーションから画面コンテンツに受け渡される。この仕組みによりサーバ側の画面コンテンツは、静的なhtmlやJavaScript、CSS等のみとなる。一般的なATM画面レベルでは、画面コンテンツのためのサーバ側ロジックを必要とせず100%キャッシュを活用可能となる。課題2のサーバ間の通信量の削減は、(2)クライアントアプリケーションの実行機能で説明した対策と合わせて、最大限にキャッシュの活用を可能としている。

図5に、次世代ATM-FWを利用した次世代ATMソフトウェアの通信イメージを示す。

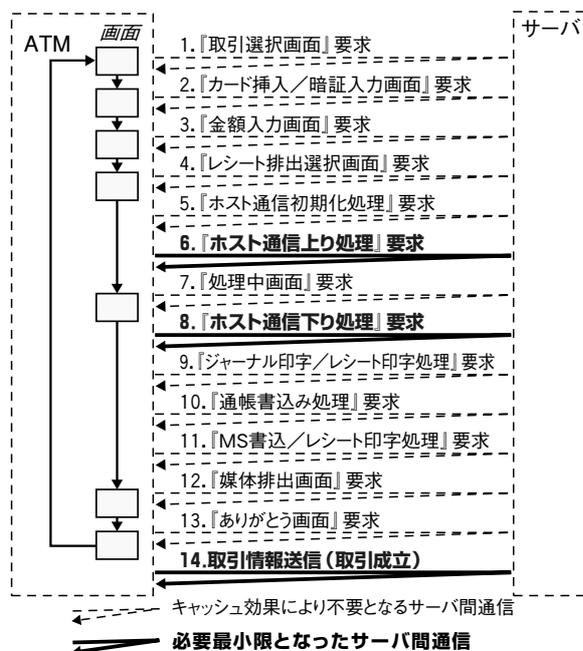


図5 次世代ATM-FWを適用時の通信イメージ

図5の例は、支払取引の例である。点線は一般的WebATMの構成における通信のイメージであり、14回の通信を必要としている。次世代ATM-FWを利用した次世代ATMソフトウェアの支払取引としては、実線に示す3回のサーバ通信で実行可能となる。

一方、課題3の対策としては、画面コンテンツが表示のみ意識すればよい非常にシンプルな構造になったことで、画面コンテンツとして複雑なATM制御を意識することなく自由に画面コンテンツを開発可能としている。ATMとしての画面表示は、ブラウザ制御機能を通して依頼されるが、実際の画面コンテンツの表示は、画面コンテンツの開発者が自由に定義できる。複雑な画面遷移を画面コンテンツ内で独自に展開する場合でも、ブラウザ制御機能からの画面表示依頼に対し適切な応答を返すことで、クライアントアプリケーション側は画面コンテンツ内の処理を意識することなくATMとしての取り引きを継続できる。このように画面コンテンツは、ブラウザ制御機能や(1)情報交換機能により必要となるクライアントアプリケーションとのI/Fを維持し、利用することでATM制御との連携を行いつつ自由なユビキタスサービスを実現することが可能となる。

(4) サーバ通信機能

サーバ通信機能は、汎用的なXMLフォーマットによる情報交換の仕組みをクライアントアプリケーションとサーバアプリケーションに提供する。これにより金融機関に異なるホスト通信の処理は、サーバアプリケーション内のカスタマイズ部に集約され、クライアントアプリケーションは、ホスト通信の差異に影響を受けずに取引機能の実現に集中することが可能となる。また、サーバ通信機能は、サーバアプリケーションから各ATM内のクライアントアプリケーションへの通信を実現するための機能を提供する。これにより、ホスト指示の電文等をATM側に伝えることを可能としている。

(5) サーバアプリケーションの実行機能

一般にモジュール間のプログラム上の依存関係を排除するために、DI (Dependency Injection) と呼ばれる技術が採用されている。DIを採用することで、プログラム上の依存関係は、設定ファイルに退避される。モジュール間の依存関係を排除することは、一方のプログラムを修正する際の、もう一方のプログラムの品質の維持を可能とする。サーバアプリケーションの実行機能は、金融機関に異なるホスト通信処理を効率よく実装するために、このDIの技術を採用している。サーバアプリケーションは、

設定ファイルの変更のみで具体的なホスト通信処理を取り入れることが可能である。サーバアプリケーションは、特定の金融機関のホスト通信処理に影響を受けない汎用的な処理として実装され、クライアントアプリケーション間の品質を維持できる。

サーバアプリケーションの実行機能は、これら機能を提供しつつも、一般的J2EE (Java 2 Enterprise Edition) の構造に素直に従っている。これにより、一般的なWebシステムとしてのサーバ側の可用性対策やセキュリティ対策などを柔軟に取り入れることが可能となっている。また、一般的なJ2EEの構造に従うことで、さまざまな一般的Webシステムと連携することが柔軟にできるとともに、課題3のサーバ側の対策ともなっている。

まとめ (今後)

次世代ATM-FWの実現により、既存のWebATMの抱える課題を解決しつつ、将来のユビキタスサービスへの対応も可能になった。今後は、ユビキタスサービスの次世代ATMへの提供をより効率的に実現するために、ユビキタスサービスとして必要とされる汎用的仕組みを体系化し次世代ATM-FWの拡張を行う。

さらに、同様なWeb系技術の基盤整備が求められる顧客操作型チャンネル全般に対しこのフレームワーク技術の展開を図っていく。 ◆◆

● 筆者紹介

森田満二 : Mitsuji Morita. システム機器カンパニー 営業本部 プロダクトSE部