

千葉 伸浩 鈴木 勝彦
NTT 情報流通プラットフォーム研究所

1. はじめに

IC カードとは、クレジットカード大のカードに CPU や RAM、不揮発性メモリなどの IC チップを搭載したもので、サーバなどからのコマンドを受信して様々な処理を行い、結果を返却する機能を持つ。初期の IC カードは、1 枚で 1 つのアプリケーション(AP)しか格納することができなかったが、1 枚の IC カードに複数の AP を搭載可能な IC カード^{[1][2]}が登場し、1 枚のカードで複数の IC カードサービスを利用できるようになった。今後は、カード内の複数 AP が互いに連携することで、より付加価値の高い IC カードサービスの実現が期待されている。本稿では、このような複数 AP 間での連携を安全かつ柔軟に行うために、制御構造を記述可能なスクリプトを提案する。また NTT 環境研の ELWIS カード^[3]と我々が開発した IC カード OS-WAOS^{-[4]}上にこのスクリプトを実装し、実用性について、処理速度や必要なハードウェアリソースの面から評価を行なう。

2. AP 間連携

AP 間連携とは、IC カードに複数の AP が格納された状態において、AP が他の AP を起動させたり、AP 間で情報の受け渡しをすることである。例えば、電子チケット AP でチケットを購入すると、電子マネー AP が起動し、これが電子チケットの金額情報を受け取って電子マネーの決済をするというような AP 間連携が考えられる。

3. 従来の技術と問題点

AP 間連携を行う上で重要なのは、以下の 2 点である。

(1) AP 間情報の安全性

AP 間でやり取りする情報は、第三者が改ざんしたり盗聴できないように、IC カード外に出さないことが望まれている。

A study of Smart Card system model which used script for cooperation between applications in Smart Card

Nobuhiro CHIBA, Katsuhiko SUZUKI

NTT Information Sharing Platform Laboratories

(2) AP 組み合わせの柔軟性

IC カードに格納されている複数の AP を様々な組み合わせることができれば、様々な IC カードサービスを実現することができる。

このような観点から、従来技術を用いた場合について以下のようなモデルを用いて検討を行った。

・サーバ経由モデル

IC カード内の AP 間連携を行う際、AP 間の情報のやり取りを、IC カードサービスサーバなどを介して行うモデル

・データ共有機能を用いたモデル

AP 間でデータの共有を行なうことで、IC カード内で AP 間の情報のやり取りを行うモデル

・Delegation 機能を用いたモデル

IC カード内の AP が、他の AP を起動したり、データのやり取りを行うモデル

・オンラインスクリプト機能を用いたモデル

サーバから IC カードへコマンドを発行する際、コマンドを複数まとめて一度に IC カードへ送るモデル

これらのモデルを用いて AP 間連携を行った場合、以下のように、従来技術では AP 間情報の安全性と AP 組み合わせの柔軟性を同時に満たすことはできない。

	AP 間情報の安全性	AP 組み合わせの柔軟性
サーバ経由	×	○
データ共有	○	×
Delegation	○	×
オンラインスクリプト	×	△

4. 制御スクリプト

前記の問題を解決するためには、IC カードサービスサーバなどで行っている、コマンドの発行や、AP 間情報の受け渡しや、AP 処理結果の評価を IC カード内で行う機能が必要であり、また、様々な AP に対して様々なコマンドを発行することが必要である。

本研究では、ICカードサービスサーバ側で行っているAP間連携のための処理を、制御構造を記述可能なスクリプト（制御スクリプト）として記述し、サーバからICカードに一括で送り、ICカード内でこの制御スクリプトを実行させるモデルを提案する。

制御スクリプトに要求される機能は、以下の通りである。

APの選択：ICカード内APの選択

コマンド発行：ICカード内APなどへコマンド発行

コマンドレスポンス処理：レスポンスの受け渡し

コマンドレスポンス評価：評価を行って処理の分岐

制御スクリプトコマンド仕様を以下に示す。

Select	APの選択
Put	APへコマンドの発行
Get	コマンドレスポンス取得
Goto	指定位置へ処理の移動
Log	レスポンスをバッファへ格納
If then	分岐処理
For next	繰り返し処理
End	バッファに格納された情報を返却して終了

図1 制御スクリプトコマンド仕様

5.実験

今まで述べてきた制御スクリプトが処理速度や必要リソースの面で、実用的であるかを検証するためにICカード上に制御スクリプト処理部を実装し、実験を行った。

5.1.実装環境

NTT環境研が開発したELWISEカード(CPU16bit, FLASHメモリ512KB)と、NTTプラットフォーム研が開発したICカードOS-WAOS-1を用いてスクリプト処理部を実装した。

5.2.測定方式

ICカードにスクリプト処理部を実装して、必要リソース(FLASHメモリ, RAM)の測定を行った。

また、サーバなどからICカードへ送るコマンド群を、制御スクリプトとしてまとめて記述し、それをスクリプト処理部が処理を行ったときの、1コマンドあたりのオーバーヘッドの測定を行った。

5.3.測定結果

スクリプト処理部を実装した結果、プログラムサイズ、RAMサイズとも、小さなサイズであり、16KB程度の記憶容量をもつICカードでも、実装できることが分かった。

また、スクリプト処理部の処理速度について測定した結果、スクリプト内の各コマンドを処理するためのオーバーヘッドは非常に小さく、サーバーICカード間でコマンドを送受信する際に生じる通信時間のオーバーヘッドと比較すると1/4以下であった。

6.考察

今回スクリプト処理部の実装を行い、プログラムサイズ、必要RAMサイズを測定した結果、中程度の記憶容量をもつICカードであれば実装できることが分かり、ハードウェアリソース的に実用的であることが確認できた。

処理速度についても、スクリプト処理のオーバーヘッドが小さいため、サーバからICカードへ複数コマンドを送信するよりも、複数コマンドをスクリプトとして記述してICカード内で実行させたほうが速く、実用的であることが確認できた。

7.終わりに

本稿では、ICカード内でAP間連携を行う際の問題点を指摘し、その解決策として制御構造を記述可能なスクリプトを提案し、実装を行い、必要リソースや処理速度の面から実用性について評価を行った。結果、どちらの面においても実用的であることが確認できた。今後は、AP間連携を行う際に発生するICカードサービスサーバ側の問題点を抽出し、その解決策について検討を行う。

参考文献

- [1]Sun Microsystems Inc : "Java Card Technology", <http://www.javasoft.com/products/javacard/>
- [2]Maesco Ltd. : "Multos", <http://www.multos.com/>
- [3]SMP(ELWISE)カードの報道発表 : <http://www.ntt.co.jp/news/news98/9810/981016a.html>
- [4]庭野、鈴木、千葉、村山、細田：“権限指向高信頼マルチアプリケーションICカードOS-WAOS-1”，電子情報通信学会 知能ソフトウェア工学研究会(1999.8).