

## Ruby on Railsを用いた 非接触ICカードシステム構築フレームワークの提案

平本優司<sup>†</sup> 山本哲也<sup>†</sup> 寺田 努<sup>‡</sup> 塚本昌彦<sup>‡</sup> 柳沢 豊<sup>\*</sup>

<sup>†</sup>神戸大学大学院自然科学研究科 <sup>‡</sup>神戸大学大学院工学研究科  
<sup>\*</sup>NTTコミュニケーション科学基礎研究所

RFIDは、流通、物品管理、履歴管理などに広く用いられるようになってきたが、未だに応用範囲は限られており、身近な小規模システムにおける導入が進んでいない。その原因の1つとしてRFIDを用いたシステムを構築するプラットフォームが整備されていないことが挙げられる。そこで、上記の問題を解決するため、RFIDを用いたシステムを容易に実現できるフレームワークを提案する。まず、RFIDタグを用いた授業支援システムおよび投票システムを実際に構築・運用し、フレームワークに必要とされるシステム要件を明らかにする。次に、それらの要件をもとにRFIDを用いたシステムを構築するためのフレームワークの設計・実装を行う。提案するフレームワークを用いて実際に在席管理システムを実装し、フレームワークの有効性を明らかにする。

### Proposal of a Framework for Constructing Contactless IC Card Systems using Ruby on Rails

Yuji HIRAMOTO<sup>†</sup> Tetsuya YAMAMOTO<sup>†</sup> Tsutomu TERADA<sup>‡</sup>  
Masahiko TSUKAMOTO<sup>‡</sup> Yutaka YANAGISAWA<sup>\*</sup>

<sup>†</sup>Graduate School of Natural Science and Technology, Kobe University <sup>‡</sup>Graduate School of Engineering Research Course, Kobe University <sup>\*</sup>NTT Communications Science Basic Laboratory

Although RFID is widely used in the area of the circulation, the inventory management, and the traceability management, these conventional systems have single purpose and it is difficult to apply the system to another purpose. One of the reasons for this problem is the lack of common platform for constructing a system with RFIDs. Therefore, we propose a new framework to construct systems with RFIDs easily. As the first step, we have implemented and actually used several RFID applications to investigate the requirements for the framework. Then, we construct a framework considering the requirements from the pilot study. Actually we implement a presence management system using our framework, and show the effectiveness of the proposed framework.

#### 1 はじめに

近年、バーコードや電子的なIDタグを使って人やモノにデータを貼り付けるデータキャリア技術が、人やモノ、場所の識別や管理のためにローコストで利用されるようになった[1]。その中でも、RFID(Radio Frequency Identification)は、情報を電子的に保持して非接触で情報交換を行うことができるため、現在最も注目されるデータキャリアの一つである。

また、コンピュータの小型化・高性能化や通信機器の発展・普及により、実世界のモノや人の活動をきっかけにして、コンピュータ情報を操作しようという、実世界指向インタフェースへのアプローチが多く行われており、ユビキタスコンピューティング環境に関する注目が高まっている。RFIDは、人やモノに情報を貼り付け検出する手法として利用されることが多く、ユビキタス化に向けて重要な基盤技術の一つとして考えられている。

一方で、RFIDはバーコードに代わるデータキャリア技術として、すでに流通、物品管理、履歴管

理など様々な用途に利用されてきている[2]。日本においては、おサイフケータイと呼ばれるFeliCaチップを搭載した携帯電話やJR東日本の「Suica」、ビットワレットの電子マネー「Edy」に代表されるFeliCaカードと呼ばれる非接触ICカードが急速に普及している。FeliCaはSonyが開発した非接触型ICカードの技術方式である。このような非接触ICカードには一つ一つ固有のID番号が振り分けられており、その固有IDを用いることで、低コストでカードの種類に限定されないサービスやアプリケーションが構築可能であるが、現在非接触ICカードは単一の目的で使用されることが多い。また、RFIDを用いた関連研究においても単一のアプリケーションに特化したRFIDシステムの提案が多く、汎用的に使えるシステムを提案しているものは少ない。その原因の一つとしてRFIDを用いたシステムを構築する際、便利なプラットフォームが整備されていないことが挙げられる。

本研究では、その問題を解決するため、RFIDを用いたシステムを容易に実現できるフレームワーク

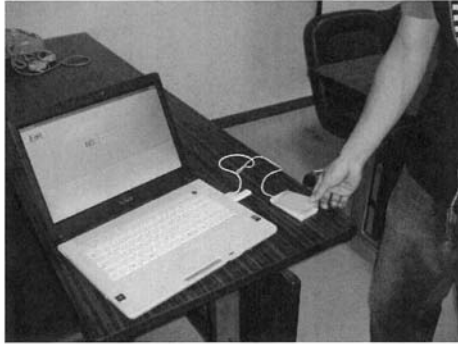


図 1: 授業支援システムの様子

を提案する。提案フレームワークを用いることで、これまで RFID が利用されてこなかった様々な場面で RFID を用いたシステムやサービスが増加することを狙う。本論文ではまず、提案するフレームワークに必要とされるシステム要件を明らかにするため、非接触 IC カードを用いた授業支援システムおよび投票システムを実際に構築し運用実験を行う。そこからいくつかのシステム要件を抽出し、それらのシステム要件を満たすフレームワークの機能を設計する。

以下、2 章では、システム要件抽出のために実装・運用実験を行った授業支援システムおよび投票システムについて述べる。3 章では、運用結果から得られたシステム要件について述べた後、提案するフレームワークの設計方針について述べる。4 章では、3 章で述べた設計方針に基づき実装した非接触 IC カードシステム構築フレームワークの実装について述べる。また、そのフレームワークを用いて構築を行った在席管理システムについて述べる。5 章でまとめと今後の課題について述べる。

## 2 非接触 IC カードアプリケーションの運用

非接触 IC カードシステムの特徴は「IC カードをかざすことで情報を入力する」という物理的なインタフェースであり、ユーザは意図的なインタラクションを必要とする。そのため、非接触 IC カードシステムは出席管理や在室管理に適していると考えられる。そこで、多くの人数を管理する必要のある授業の出席管理のためのアプリケーションを実装し、運用を行った。さらに、授業支援システムでの結果を踏まえて多数のリーダを連携させた投票システムについても実装および運用を行った。

表 1: 授業支援システム運用状況

科目名	期間	受講者	被験者
情報の世界	2006 年後	152 人	146 人
計算機工学 I	2007 年前	101 人	71 人
ソフトウェア構成特論	2007 年前	96 人	82 人

### 2.1 授業支援システム

従来の授業では、口頭や紙媒体による出席管理が主に用いられていた。近年、様々な形で授業の出席管理およびサポートを行う研究が行われている。授業中に表示した QR コードを学生が携帯電話で認識させることで出席管理を行うといった研究や、学生証を非接触 IC カードにして出席管理に利用するといった大学も増加している [3]。本研究で構築した授業支援システムでは、FeliCa チップを搭載した非接触 IC カードまたはおサイフケータイを受講している学生に用意してもらいシステムの運用を行った。非接触 IC カードまたはおサイフケータイを教室の入り口に設置した IC カードリーダ・ライタにかざすことによって授業の出席管理を行い、その出席データを利用してメール送信およびメッセージの表示を行うシステムである。メール内容は、授業の出席回数、遅刻回数、講義の資料など多岐にわたる。図 1 に授業支援システムを利用している様子を示す。

#### 2.1.1 システム構成

このシステムでは、IC カードリーダ・ライタとして Sony の PaSoRi(RC-320) を使用した。また、データベース管理システムには MySQL、システム開発には、Sony の SDK for FeliCa Lite Ver.2.0 を使用し、Microsoft Visual C++ 2005 を用いて実装を行った。

#### 2.1.2 評価

このシステムを神戸大学において開講されている授業科目「情報の世界」および「計算機工学 I」、「ソフトウェア構成特論」の講義において運用した。受講者には非接触 IC カードまたはおサイフケータイを各自用意してもらい被験者としてシステムを使用してもらった。また、非接触 IC カードまたは携帯電話が用意できなかった受講者や IC カードを忘れた受講者に対しては、紙媒体で出席管理を行った。表 1 に授業支援システムを運用した期間、受講者数、被験者数を示す。

このシステムでは、一台のノート PC と一つのリーダ・ライタを使って運用していたため、一度に一枚のカードしか読み取れず出席登録待ちの行列ができるといった問題があった。一方、従来の口頭や紙媒体での出席管理方法と比較して、下記の特徴をもつ。



図 2: 投票システムを利用する様子

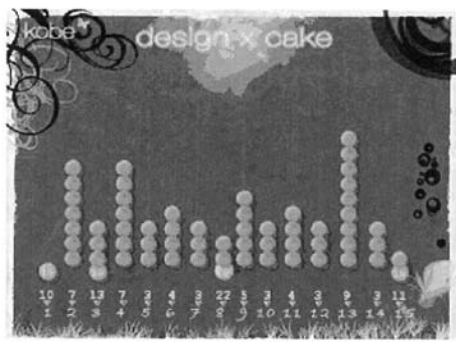


図 3: 投票結果表示画面

- データ管理が容易である
- 出席時間および退席時間を認識することができるため遅刻の判別が可能
- メール配信により講義の資料配布, アンケートなどを容易に行える

また、ユーザの出席データによって送信するメールの内容や表示するメッセージの内容などの変更を行った。このシステムを応用することで、店舗における広告配信やポイントシステムなどが実現できる。しかし、このシステムは C++ で実装したため、細かな変更を行う度にコンパイルしなおさなければならないという問題や表示画面のデザイン変更が簡単に行えないなどの問題があった。また、データベース関連の処理において、毎回 SQL 文を記述しなければならず、メール内容などの変更を柔軟に行うことができなかった。

## 2.2 投票システム

この投票システムでは、複数の作品の中から一つもしくは複数個気に入ったものを選ぶような投

票において、非接触 IC カードまたはおサイフケータイをリーダー・ライタにかざすことによって投票結果をリアルタイムに表示する。この投票システムは、2007 年 9 月 9 日に神戸市などにより主催された「KOBE CAKE DESIGN FESTA 2007」において実際に使用し、96 名の参加者に投票を行ってもらった。その際には、一人につき一回の投票権利を有するとし、何度でも投票をし直すことができる仕様にした。図 2 に投票システムを利用する様子、図 3 に投票結果表示画面を示す。

### 2.2.1 システム構成

「KOBE CAKE DESIGN FESTA 2007」では、15 個の作品があり、各作品のブースごとにリーダー・ライタを設置した。ブースごとの距離が離れていたことから 4 台のノート PC に 15 個のリーダー・ライタを接続し、投票結果を表示するスクリーン用に 1 台のノート PC を使用した。また、非接触 IC カードを持っていない投票者のために 100 枚の非接触 IC カードを用意した。投票結果表示画面は、Flash を用いて作成した。また、このシステムの別の運用例として、神戸大学大学院工学研究科塚本研究室における研究紹介デモンストレーションにおいて、各デモンストレーションの評価を順位付けするためにこの投票システムを使用した。その際には、一人につき何度でも投票できる仕様にした。

### 2.2.2 評価

この投票システムによって、非接触 IC カードを複数台連携させて投票に用いるといった新しい RFID の使い方を提案した。また、本システムの利点としては、下記のもの挙げられる。

- 集計の手間削減
- ログ解析によりユーザの動向を解析可能

このように投票において RFID を用いることで、投票回数などを自由に変更でき、ID 情報のログを解析することでユーザの好みや行動の様子などを解析することが可能である。しかし、このシステムでは投票対象の数や投票者の数などが変更された場合には、データベースのスキーマやデータの表示方法などを変更しなければならないという問題があった。

## 3 非接触 IC カードシステム構築フレームワークの設計

前章で述べたシステムの実装・運用実験から得られた要件を整理し、提案するフレームワークの設計方針について述べる。

### 3.1 フレームワークの要件

これまでに述べたように、非接触 IC カードを用いたシステムやアプリケーションは特定用途には現状でも広く用いられているが、汎用的システム基盤

が存在していないため、手軽に様々な目的のためのアプリケーションを構築することが難しい。つまり、システム構築を容易に行えるようなプラットフォームを構築すれば、より多くの用途で非接触 IC カードを用いたサービスやシステムが使われるようになると考えられる。2 章で述べたように、実際には、非接触 IC カードの固有 ID を用いれば、小規模ながら低コストで容易に非接触 IC カードを用いたシステムが実現できる。

本章では、このような非接触 IC カードシステムの汎用プラットフォームを構築する。前章で述べたように、非接触 IC カードを用いた授業支援システムおよび投票システムの実装・運用から、以下のようない知見が得られた。

- 低コストでシステムが実装可能
- 非接触 IC カードとメール配信を組み合わせたシステムは汎用的に利用できる
- メール配信やメッセージ表示などを行う際、データベースとのデータのやりとりが多い
- 部分的な機能の入れ替えやシステム稼働中のシステム変更が容易に行なえることが必要
- システム構築を容易に実現できるプログラミングモデルが必要

これらの知見を活かし、問題点を克服するようなフレームワークを考えなければならない。そこで、フレームワークの要件として、(1) データベース処理の単純化、(2) システム構築を容易に実現できるプログラミングモデルの 2 つに着目した。以下では、これらの要件について説明する。

- (1) **データベース処理の単純化** 非接触 IC カードを用いたシステムでは、データ表示またはデータ配信などのデータベース処理が多い。データベースからデータを取得する際には、SQL によってデータを取得することが一般的であるが、この処理には専門的な知識を要するため、できるだけデータベース処理を記述する負荷を削減する必要がある。
- (2) **容易なプログラミングモデル** 非接触 IC カードを用いたシステムは、リーダ・ライタにカードをかざしたことをきっかけとして動作を実行するといった使い方が多い。そのようなきっかけと状況、動作を定義するだけでシステム構築が可能なプログラミングモデルの採用が有効である。

### 3.2 フレームワークの設計方針

さきほど述べた必要要件に基づきフレームワークの設計方針について述べる。

表 2: イベント一覧

イベント	説明
READ.ID	ID 読み取り
RELEASE.ID	ID 読み取り終了
TIMER	タイマ発火
CLICK.BUTTON	ボタンをクリック
DATABASE.INSERT	データの挿入
DATABASE.DELETE	データの削除
DATABASE.UPDATE	データの更新
DATABASE.SELECT	データの取得

表 3: コンディショナー一覧

コンディショナー	説明
TAG.ID	タグの ID
READER.ID	リーダの ID
BUTTON.ID	ボタンの ID
TIMER.ID	タイマの ID
DATABASE	データベースの値
TARGET	対象テーブル
NEW	更新されたデータの値
OLD	更新前のデータの値

- (1) **データベース処理の単純化** 非接触 IC カードシステムでは、固有 ID に関連したデータベーステーブルを複数用いることが多い。そこで、データベース上からデータの取得を単純化するためにオブジェクト・リレーショナル・マッピング (ORM) を用いる。ORM ではデータベースのレコードがオブジェクトとして直観的に扱えるようになり、データベースにアクセスするプログラムを記述する処理が容易になる。この ORM は既存の WEB アプリケーション開発フレームワークではよく用いられており、その代表的なフレームワークの 1 つである Ruby on Rails では ActiveRecord (AR) で ORM を実現している。そこで提案するフレームワークでは Ruby on Rails を採用する。
- (2) **容易なプログラミングモデル** あるイベントをきっかけにしてその時の状況に応じて動作を実行するといったモデルには、イベント駆動型のプログラミングモデルが適している。そこで、このフレームワークでは、アクティブデータベースで用いられている ECA ルール機構を採用する [4][5]。ECA ルールとは発生する事象 (Event)、条件 (Condition)、実行内容 (Action) の 3 つを 1 組とした動作定義言語であり、ルールを組み合わせることでシステム容易に構築できる。

表 4: アクション例

アクション	説明
SEND_MAIL	メール送信
JUMP_URL	指定した URL へ移動
SET_TIMER	タイマ開始
KILL_TIMER	タイマ解除
SHOW_DATA	データ表示
DATABASE	データベース操作

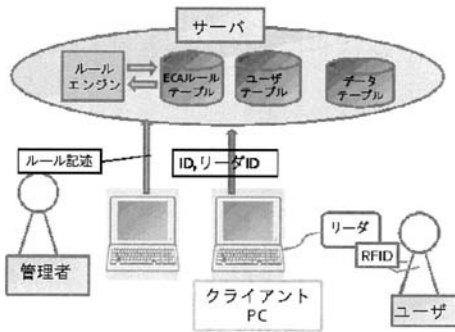


図 4: システム構成

### 3.3 ECA ルールの言語仕様

システムの運用実験から得られた知見から、表 2, 3, 4 のようなイベント、コンディション、アクションを実装した。新しく変数や関数を定義することにより、自由にコンディションやアクションを追加できる。

### 3.4 フレームワークの構成

提案するフレームワークを用いたシステムの構成を図 4 に示す。システムの動作の流れを以下に示す。

1. ユーザがリーダー・ライタにカードをかざす
2. クライアントシステムがタグ ID, リーダ ID をサーバのデータテーブルに格納
3. ルールエンジンがテーブル更新をイベントとして検知
4. ECA ルールテーブルに格納されているルールの中で発生したイベントに適合するものを抽出
5. 抽出したそれぞれのルールに対してコンディションを評価し、満たされればアクション実行

ECA ルールの記述は、ECA ルールテーブルに用意されたもしくは定義したイベント、コンディション、アクションを格納することで行う。ECA ルールエンジンはデータテーブルとユーザテーブルの変

更を常に監視しており、それらの変更を検出すると ECA ルールテーブルにあるルールの中で、発生したイベントに適合するものを抽出し、コンディション評価およびアクション実行を行う。

## 4 システムの実装

これまでに述べた非接触 IC カードシステム構築フレームワークを実装した。実装には Ruby を使い、データベース管理システムには MySQL を用いた。カードリーダーには PaSoRi(RC-320)、RFID カード規格としては FeliCa チップを搭載している非接触 IC カード及びおサイフケータイに対応した。ただし、タグ ID を読み取るプログラムには、Sony の SDK for FeliCa Lite Ver.2.0 を使用し、Microsoft Visual C++ 2005 を用いて実装した。

### 4.1 Ruby on Rails

Ruby on Rails は WEB アプリケーション開発のフレームワークである。フレームワークの要件においてデータベース処理の単純化を挙げたが、Ruby on Rails では、AR という ORM ライブラリが用意されており、これを利用して簡単にデータベース上の情報を Ruby オブジェクトとして扱うことができる。

また、本フレームワークでは、メッセージ表示などのデータ表示を WEB ブラウザで行うことを想定している。WEB アプリケーション開発用のフレームワークである Ruby on Rails を用いることで開発期間を短縮できる。さらに、ECA ルールのアクションを追加する際には、Ruby on Rails のライブラリを利用できるようにしており、様々なアクションが容易に実装できる。

### 4.2 アプリケーション構築例

提案するフレームワークを用いた簡単なアプリケーション構築例として在席管理システムを構築した。本システムでは、初期設定として、各ユーザに対して非接触 IC カードのタグ ID とカードリーダーのリーダー ID、メールアドレスをあらかじめ関連付けたテーブルを用意した。本システムの主な機能を次に示す。

1. IC カードをカードリーダー上に置いている間は、タグ ID と関連付けされているユーザの在席状況を在席中にする。
2. 60 秒以上 IC カードをカードリーダーにかざさないでおくと、ユーザの在席状況を退席中にする。
3. リーダ ID と関連付けされていないタグ ID を読み取ると、そのタグ ID と関連付けされたユーザのもとにリーダー ID に関連付けされたユーザの在席状況が退席中から在席中になった時に、到着を知らせるメールを送信する。



表 5: 在席管理システムで用いた ECA ルール

EVENT	CONDITION	ACTION
Rule(1)		
READ_ID	DATABASE.USER('ID'=TAG_ID) .STATE== 'AWAY' DATABASE.USER('ID'=TAG_ID) .READER== READER_ID	DATABASE.USER('ID'=TAG_ID).STATE = 'PRESENCE' SET_TIMER(TAG_ID, 60)
Rule(2)		
TIMER		DATABASE.USER('ID'=TIMER_ID).STATE == 'AWAY'
Rule(3)		
READ_ID	DATABASE.USER('ID'=TAG_ID) .STATE== 'AWAY' DATABASE.USER('ID'=TAG_ID) .READER != READER_ID	DATABASE.USER('READER'=READER_ID) .WAITED = TRUE DATABASE.USER('READER'=READER_ID) .WAITEDID = TAG_ID
Rule(4)		
DATABASE.UPDATE	TARGET == USER NEW.WAITED == TRUE	NEW.WAITED == FALSE SEND_MAIL( DATABASE.USER('ID'=NEW.WAITEDID) .MAILADDR, NEW.NAME "さんが来ました")
Rule(5)		
CLICK.BUTTON	BUTTON.ID == 'SENDMAIL'	SEND_MAIL( DATABASE.USER('STATE'=PRESENCE) .MAILADDR, "今日あなたは出席扱いとなりました")

4. ブラウザ上のメール送信ボタンを押すと、在席状況が在席中のユーザにメールを送信する。

また、上記の機能を実際に記述した ECA ルールの例を表 5 に示す。Rule1 はリーダに IC カードが置かれた際に、その IC カードのユーザと対応する席のリーダであれば状態を出席に更新するルールである。さらに、60 秒のタイマを設定することで、60 秒間 IC カードが離れていたら (60 秒間の間に再び READ\_ID イベントが起こらなければ)、Rule2 が発火してユーザを欠席状態にする。Rule3 はあるユーザが他人のリーダのところにカードを置いたときに、そのリーダに本来関連付けられているユーザの状態に、待たれていることおよび誰から待たれているかの情報を登録するルールである。Rule4 は、Rule3 で登録されたユーザが着席した際に、登録したユーザにメールで知らせるルールである。また、Rule5 は、管理画面のウェブサイトにてメール送信ボタンを押した際に、そのときに出席状態になっているユーザにメールを送るルールである。

## 5 おわりに

本研究では、Ruby on Rails と ECA ルールメカニズムを用いて非接触 IC カードシステム構築フレームワークの実装を行った。また、そのフレームワークを用いたアプリケーション構築例として在室管理システムを実装した。今後、さらに様々なシステムに利用するためには、汎用性の高いコンディションやアクションを追加していく必要がある。また、現状のシステムでは ECA ルールによりシステム構築の負荷を下げてはいるが、アクション記述等を実際に行うには、ある程度プログラミングの知識が必要であるため、プログラミングの知識がない人でも容易にアプリケーション構築が可能な方式を考える必要がある。その例として、GUI を用いて必要なものを組み合わせるといった方式が考えられる。

## 参考文献

- [1] 椎尾一郎, 早坂 達: モノに情報を貼り付ける, 情報処理, Vol. 40, No. 8, pp. 846-850, 1999.
- [2] B. Nath, F. Rynolds and R. Want: RFID Technology and Applications, Pervasive Computing, pp. 22-24, 2006.
- [3] 樋川和伸, 岡田正則, 中西一夫: IC カードを利用した携帯ユビキタス授業支援システム, 教育システム情報学会第 30 回全国大会講演論文集, pp. 373-374, 2006.
- [4] 寺田 努, 塚本昌彦, 坂根 裕, 義久智樹, 岸野泰恵, 早川敬介, 柏谷 篤, 西尾章治郎: ユビキタスコンピューティングのための入出力制御デバイスの動作記述言語, 第 1 回情報科学技術フォーラム (FIT2002) 論文集第 4 分冊, pp. 197-198, 2002.
- [5] 石川 博: アクティブデータベース, 情報処理, Vol. 35, No. 2, pp. 120-129, 1994.