2004－ＭＢＬ－29　（６）
2004／5／13

アクティブ型RFIDを用いて場所に応じた情報を配信するシステムのためのAPIの設計と実装

谷　隆三郎†　橋本　和樹†　須子　善彦†　南　政樹†　村井　純†

† 慶應義塾大学 〒252-0816 神奈川県藤沢市遠藤 5322
E-mail: †{tani,kazuki,scottie,minami,jun}@sfc.wide.ad.jp

あらまし　アクティブ型RFIDを用いて実空間における位置情報を取得できる。そして位置情報を利用することで利用者により質の高い情報を配信するシステムが構築できる。しかしこれらのシステムは独自の設計であるため、新たにアプリケーションを開発する際において大きな障害となっている。そこで本研究ではこれらのアプリケーション開発を容易にするためにアクティブ型RFIDシステムから位置情報をアプリケーションへ提供するためのAPIの設計と実装を行う。

キーワード　RFID, アクティブタグ, API, IPv6, 情報管理システム

A design and implementation of API for the system which distributes the information according to the place using active type RFID

Ryuzaburo TANI†, Kazuki HASHIMOTO†, Yoshihiko SUKO†, Masaki MINAMI†, and Jun MURAI†

† Keio University 5322 Endo, Fujisawa-shi, Kanagawa, 252-0816 Japan
E-mail: †{tani,kazuki,scottie,minami,jun}@sfc.wide.ad.jp

Abstract　If active type RFID is used, it is possible to acquire people's position information in real space. Real space information is acquired using this technology, and the system which distributes high quality information to users exists. However, since these systems are built using the original information management system, in case they newly develop application, they have been serious obstacles. In this research, in order to make these application development easy, a design and implementation of API for offering real space information from an active type RFID system to application are performed.

Key words　RFID, Active tag, API, IPv6, Infomation Management System

1. はじめに

利用者に最適な情報を提供するためにはまず利用者の個人を識別しなければならない。利用者にIDを割り当てて個人を識別する技術にはバーコードやICカードといった技術が存在するが、どちらも利用者が意識してIDを読み取らせなければならない。

そこでバッテリーを内蔵し、一定の周期でIDを発信することにより長距離での読み取りが可能なアクティブタグを用いる。このような技術は今後必要とされ、より優れた自動認識技術になると考えられる。そこで本稿では、アクティブタグを用いたシステムから位置情報を取得するためのAPIを提案する。

2. 場所に応じた情報配信システム

場所に応じた情報配信システムを構築するには以下の機能が必要である。

- 利用者を識別するためのIDを取得
- 利用者の位置情報を取得

本章ではこれらの機能の役割と機能要件について述べる。

2.1 利用者を識別するID

実空間における場所に応じた情報配信システムを実現するためには、まず全ての人に対してユニークなIDを割り当てて利用者を識別することが必要である。これは任意のユーザに対して情報を送信する場合において、誰に送るのかを決定するために必要な情報である。

これまでの個人を特定してサービスを提供するシステムでは、メールアドレスなどを ID として利用者を識別している。しかし実空間では、その場でメールアドレスを入力させるといったことが非常に困難であるため、それに代わる ID 及びその ID を自動で認識する必要がある。この ID は利用者が意識して操作することなくシステムに認識させることができ、その ID から個人を特定できることが求められる。そしてこれらの ID と個人情報を結びつけることで情報を配信する際の送信先や内容などを変化させることが可能になる。

2.2 利用者の位置情報

場所によって異なる情報を配信するためには利用者の現在位置を取得することが必要である。この情報は前節で述べた ID と結びつけることで誰がどこにいるのかがわかるようになる。

位置情報は www の参照履歴と同じような役割を果たすことができる。例えばこれまでに訪れた場所の履歴を参照することで、その人が同じ場所に何度訪れたかなどがわかる。これによりその人が何に興味があるのかを推測することで、その人とって価値のある情報を提供することが可能になる。また人だけでなく物にもタグをつけておくことでその場所に関して充分な情報がない場合にも、何が存在しているのかということからその場所の特徴を推測することも可能である。

3. アクティブタグを用いたシステム

アクティブタグとはバッテリーを内蔵した小型無線チップである。このチップは一定の周期で ID を発信しており、リーダと呼ばれる外部機器によってその ID を取得することができる。アクティブタグの特徴はバッテリーを内蔵していることで、バッテリーを持たない場合よりも長距離通信が可能である。本章ではこのアクティブタグが前章で述べた機能に対してどのような点で適しているのかを述べる。

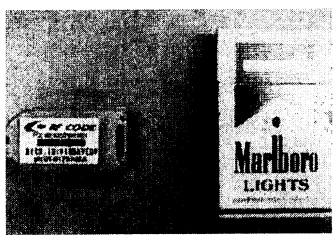

図 1 アクティブタグ

3.1 ユーザの識別

アクティブタグは一定の周期で ID を発信している。その ID をリーダが受信することで、タグがリーダの読み取り範囲内へ入ってきたことを認識できる。タグの ID と個人情報が結び付けられている場合には誰がその範囲へ入ってきたのかということがわかる。またリーダは読み取り範囲が半径 20m 程で、アッテネータを使用することで検出範囲を調整することもできる。

これらの特徴から利用者にアクティブタグを持たせることで簡単に ID を割り当てることが可能である。また長距離通信が

図 2 リ ー ダ

可能であることから、利用者がリーダの読み取り範囲内に入るだけで ID を認識することができる。こうして実空間上において自動的に利用者を特定することが可能である。

3.2 ユーザの位置情報

場所に応じた情報を配信するには利用者の現在位置を知らなければならない。リーダが 1 台のみ、もしくは全てのリーダを同じ場所だと想定してサービスを提供するならば、どのリーダから検出されたかを考慮する必要はない。しかし、リーダが複数台設置されていてそれぞれの場所で異なる内容の情報を配信する場合には、どのリーダで検出されたのかという情報が必要である。そこでリーダにも ID を割り当てて検出された ID に付加することで、どのリーダから検出された ID なのかを判断している。

この方法は利用しているアクティブタグ及びリーダによって異なる手段で実現されている。例えばネットワークに対応しているリーダの場合には MAC アドレスを ID として利用しているものが存在する。その他には、制御用プログラムがあらかじめ決められた ID を付加するといったようなリーダも存在する。こうして得られたリーダ ID からそのリーダが設置されている場所の情報を取得することができる。

ここで得られる位置情報はリーダの読み取り範囲で区切られたエリアごとの情報である。より精度を上げたい場合にはリーダの読み取り範囲を狭くしてその分リーダの数を増やすことで可能になるが、現実的にそれを実現するにはコストやスペースの問題から難しいと思われる。そこで通常は部屋に一つのリーダを設置して、部屋単位での位置情報を取得できるような使い方を想定している。

4. システムの例

本章ではこれまでに開発したアクティブタグを利用したシステムの例を挙げる。

4.1 同窓会における再会支援システム

このシステムは、自分のいる場所へ同じ属性を持った人がくるとメールで知らせるというシステムである。

4.1.1 想定環境

このシステムは、同窓会などで過去に同じサークル、研究会、団体などに所属していた人たちが集まるといったような、同じ属性をもつ人たちが多く集まるイベントを想定している。シス

テムは以下の情報があらかじめ登録されていることを前提にしており、利用者には当日受付で渡されるタグを持って会場を歩いてもらう。

- 過去に所属していたサークル、研究会、団体
- 携帯電話のメールアドレス

4.1.2 概　　要

同じ属性を持った人が近くにいることをメールで知らせることができる。また、リーダの読み取り範囲内に存在しているタグの属性を知ることで、その場所がどういう状況なのかを知ることができる。例えば同じサークルの属性を持った人ばかりが集まっている場所ではそのサークルのOB,OG会をやっている、などの状況をある程度把握することが可能になる。

システムの構成は以下のようになっている。

- 実空間情報取得モジュール

リーダから送られてくる情報を受け取りタグの検出や退出などを判断する。また検出したタグの位置情報を実空間情報データベースに格納する。

- 実空間情報データーベース

タグのIDと結び付けられた位置情報及びその他の個人情報を管理する。

- Matching Engine

ルールに基いた条件で検索を行い、一致したIDのメールアドレスに対して情報を送信するモジュール。

新しく検出されたIDを元に、そのIDの所属を実空間情報データベースから取得する。そして所属が同じかつ現在位置が同じというルールに一致するIDを検索する。一致するIDが見つかった場合には、それらのIDからメールアドレスを取得して情報を送信する。

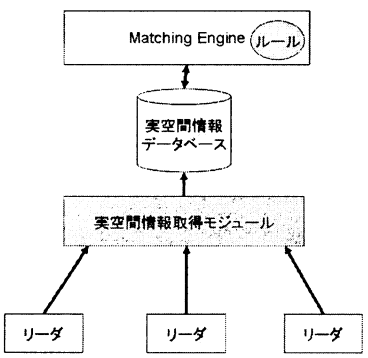

図3　同窓会における再会支援システム

4.2 待ち合わせシステム

このシステムは、自分の空き時間になると同じ時間帯で空いている友人を探し出してその人の位置情報を通知するというシステムである。

4.2.1 想定環境

このシステムは、友人同士が同じ建物内といった比較的近い場所で別々に行動している場合を想定している。例えば学校内で食事に行く仲間を探すときなどに効果的に利用することができる。

システムにはあらかじめ以下の情報が登録されていることを前提にしている。またここではタグを常に持ち歩いてるという前提である。

- 自分の空いている時間
- 知らせたい友人の情報
- 携帯電話のメールアドレス

4.2.2 概　　要

空き時間を有効に利用するために友人の空き時間を知らせるとともに相手のいる場所も知ることができる。他のスケジュール管理システムと連携することで新しい機能を提供するといった使い方を想定している。

システムの構成は以下のようになっている。

- 実空間情報取得モジュール

リーダから送られてくる情報を受け取りタグの検出・退出を判断する。またそれらの情報を実空間情報データベースへと格納する。

- 実空間情報データベース

タグのIDと結び付けられた位置情報、友人情報、スケジューリング情報を管理する。

- Task Manager

登録されているスケジューリング情報を元に検索パラメータを設定する。設定された時間になるとルールに基いた条件での検索を実行する。

- Matching Engine

ルールに基いた条件で検索を行い、一致したIDのメールアドレスに対して情報を送信するモジュール。

指定されたIDを元に友人情報及びその人の空き時間を取得する。検索実行時が空き時間に設定されているというルールに一致するIDを検索する。ルールに一致するIDが見つかった場合にはそのIDのからメールアドレス及び位置情報を取得してメールを送信する。

5. APIの提案

前章で述べたシステムはどちらもアクティブタグを利用して取得した位置情報を利用したシステムである。本章ではこれらのシステムにみられるいくつかの共通部分に関して述べ、それらの機能をアクティブタグを用いた情報配信システムにおけるAPIとして提案する。

5.1 共通する機能

前章で述べたシステムにおいてどちらも共通して持っている機能は実空間情報取得モジュールと実空間情報データベースである。実空間情報取得モジュールはリーダから得られる情報を受け取り、タグの入退出を判断して実空間情報データベースへと格納するという機能を持っている。実空間情報データベースはアプリケーションが必要とする情報を全て保持しているというモデルになっている。以下にそれぞれのシステムが保持している主な情報を示す。

実空間情報データベースの保持している情報は大きく分けて

再会支援システム	待ち合わせシステム
タグ ID	タグ ID
場所	場所
名前	名前
メールアドレス	メールアドレス
所属	空き時間
	友人情報

表 1　保持している情報

2 つの部分から構成されており、一つはリーダから得られる位置情報を保持する部分、もう一つはタグの ID と結び付けられた個人情報などを扱う部分になっている。このうち個人情報に関してはアプリケーションが提供する情報によって内容が変化するためアプリケーション独自に管理する必要があるが、位置情報に関してはどちらも同じ内容であるために共通の情報として扱うことが可能である。また位置情報を取得し、更新する役割は実空間情報取得モジュールが行っているのでこの実空間情報取得モジュールが持つ機能も全て共有することが可能である。

5.2　タグの位置情報に関する機能

前節ではそれぞれのシステムが位置情報を扱う機能に関して共有できることについて述べたが、本節では位置情報を取得する際の機能について述べる。実空間情報データベースから情報を取得する際の機能は以下の通りである。

- 再会支援システム
  - 新しく検出されたタグの ID を受け取る
  - タグが検出された場所を調べる
  - 同じ場所で検出されているタグ ID を取得する
  - 同じ所属かどうかを調べる
- 待ち合わせシステム
  - 友人リストから友人のタグ ID を取得する
  - 全ての ID に対して空いている時間を調べる
  - 一致した ID の場所を取得する

これらの機能の中で位置情報を取得する際の機能は以下に示す通りである。

- 新たに検出されたタグの通知
- ID から場所を取得
- 場所から検出されている ID を取得

これらの機能は共通の情報を用いていることから、共通のインターフェースを用意することで取得することが可能であると考えられる。

5.3　API の役割

本章でこれまで述べたことをまとめると、場所に応じた情報配信システムにおいてタグの位置情報は共通して利用できる。そしてその情報を取得するには共通のインターフェース、共通の機能を用いることができる。そこで本節ではアクティブタグを用いたシステムから位置情報を取得する際の API を提案する。

これまでのシステムでは、全ての情報は実空間情報データベースで管理されていたためにアプリケーションを開発するにはデータベースの構成を知っている必要があった。しかし、実空間情報データベースが保持している情報を位置情報と個人情報の 2 つに分け、位置情報を取得する際には API を用いることで複雑な位置情報を取得する際の手続きを隠蔽することができる。

またそうすることで位置情報を取得する実空間情報取得モジュールに変更を加える必要が出てきた場合にも、アプリケーションに及ぼす影響を最小限にすることができる。

さらにリーダの管理者ではない人たちにもリーダから得られる位置情報を提供することができる。これは API を用いてシステムの構成を隠蔽することにより、アプリケーション開発者は実空間情報取得モジュールが動作しているサーバから簡単に位置情報を手に入れることができるからである。

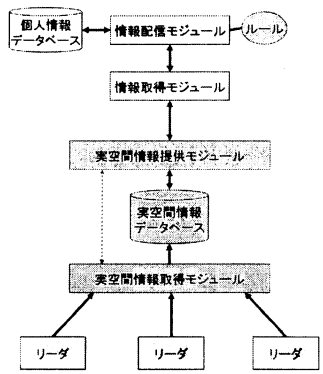

図 4　システムモデル

6. 設　　計

場所に応じた情報配信システムにおける API に必要な機能は以下の通りである。

- ID から位置情報を取得
- 位置情報から検出されている ID を取得
- 新たに検出されたタグの通知

本章ではこれらの機能を提供する C 言語 API の設計について述べる。

6.1　データ型

この API によって提供される機能は先に述べた通りであるが、これらの機能を実現するために以下のようなデータ型を定義する。

- RF_TAGID
- RF_LOCATION

RF_TAGID は複数のタグ ID を格納するためのデータ型である。これは位置情報から検出されている ID を取得する機能を実現した場合、返り値となる ID は複数になることが予想されるからである。

RF_LOCATION は複数の位置情報を格納するためのデータ型である。これはアプリケーションが現在稼動しているリーダを全て知りたいといったような要求があった場合に、複数の位置情報が返り値となることが予想されるからである。またこの

データ型を用いることで一度に複数のリーダから検出されているIDを取得したいといった要求があった場合にも対応することが可能である。

6.2 データ型に関する機能

APIの提供する関数は大きく分けて2種類に分類することができる。一つは返り値にRF_TAGID型のIDを返す機能を持った関数であり、もう一つは返り値にRF_LOCATION型の位置情報を返す機能を持った関数である。どちらも複数の値を持っている可能性があるので、get_size()関数を用意することで返り値に含まれている値の数を得ることができるようにする。またfetch_tagid()関数及びfetch_location()関数を用いてそれぞれのデータ型に含まれている値を取り出すことができる。

RF_LOCATION型は複数の位置情報を格納できるデータ型であり、このデータ型を利用して複数のリーダから検出されているIDを一度に取得することが可能である。これらの機能のためにRF_LOCATIONにデータを追加及び削除する機能を提供することができるようにする。

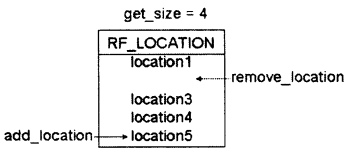

図5 RF_LOCATIONへの処理

6.3 通信インターフェース

このAPIでは情報を配信するアプリケーションとタグの位置情報を管理しているシステムが分かれているため、両者間で通信を行う場合にはネットワークを利用することを想定している。データを送信する場合には以下の手順で送信される。

- 送信側はデータ型を通知する
- 受信側がAckを返す
- 送信側はデータ型に含まれる値の数を通知する
- 受信側はメモリを確保しAckを返す
- 送信側は通知した数の値を送信する

7. 実　　装

前章で述べた3つの機能を中心に、それらの機能を実現するためのAPIの実装について述べる。

7.1 ネットワーク機能

位置情報やIDは別のマシン上で管理されており、ネットワークを利用して取得することを想定している。本来ならばどのようなインターフェースを用いても通信可能であるべきかもしれないが、ここではネットワークを前提にBSDのsocketAPIを用いて実装している。

そのためタグの位置情報を取得したり、位置情報からIDを取得するような関数を実装する際には、引数にソケットディスクリプタが渡されることとする。渡されるソケットは既にオープンされていることを想定しており、関数内ではこのソケットを利用して通信が行われる。処理が終了してもソケットはそのままの状態で、アプリケーションが明示的にクローズするまでオープンしたままである。

7.2 IDから位置情報を取得

ある特定のIDからそのタグが検出されている位置情報を取得するための関数を以下のように定義する。

get_location_from_id(int socket, char *id);

この機能を提供するには引数にソケットディスクリプタとIDを受け取り、返り値はRF_LOCATION型ポインタ変数である。これは複数の位置情報が返ってくる場合が考えられるからである。

サーバ側では受け取ったIDを用いて、実空間情報データベースシステムからそのIDが現在検出されている位置情報を取得して返す。もし検出されていない場合にはNULLを返す。

また特定のIDを指定せずに現在稼動している全ての位置情報を受け取るための関数を以下のように実装した。

get_location_all(int socket);

サーバ側ではIDを受け取らずに実空間情報データベースから全ての位置情報を検索し、その結果を返す。

7.3 位置情報から検出されているIDを取得

位置情報から検出されているIDを全て取得するための関数を以下のように定義する。

get_id_from_location(int socket, RF_LOCATION *locations);

この機能を提供するには引数にソケットディスクリプタとRF_LOCATION型ポインタ変数を受け取り、返り値はRF_TAGID型である。これはほとんどの場合において複数の値が返ってくると考えられる。また引数にRF_LOCATION型ポインタ変数を渡すのは一度に複数の場所を指定できるようにするためである。

サーバ側では受け取ったRF_LOCATION型に含まれる全ての位置情報から検出されているIDを検索し、その結果を返す。

RF_LOCATIONに新しい位置情報を追加、削除するための関数を以下のように定義する。

add_location(char *location, RF_LOCATION *locations);

remove_location(char *location, RF_LOCATION *locations);

返り値はどちらも追加もしくは削除された後のRF_LOCATION型ポインタ変数である。

7.4 新たに検出されたタグの通知

この機能は少し特殊で1回で必要な情報を取得してくるのではなく、継続的に送信してもらうように設定するといった処理を行う。この機能を実現するための関数を以下のように定義した。

set_exception(int socket);

引数にソケットディスクリプタを受け取り、返り値はint型で成功すると0を返す。この関数が実行されると渡されたコネクションに今後継続的に情報を送信してもらうよう設定する。設定が成功すると成功を表す0を返し、これ以降はunset_exception(int socket)が実行されるかコネクションがクローズされるまでこのソケットに対してタグの検出、退出が送信さ

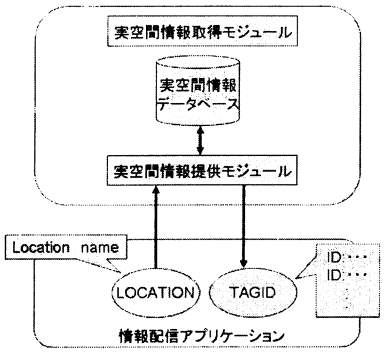

図 6　検出されている ID を取得する場合

れ続ける。

サーバ側では実空間情報取得モジュールへ直接接続し、新しいタグが検出されるかタグが退出したと判断された場合にその ID 情報を受け取り、それを返すといった仕組みで実装されている。

8. ま と め

本稿では場所に応じた情報配信システムを実現するためにアクティブタグを利用したシステムの例をいくつか挙げた。またそれらのシステムが共通して持っている情報及び、情報を取得するための機能について述べ、場所に応じた情報配信システムのための API を提案した。この API を利用することでアクティブタグを用いてタグの位置情報を取得するための共通インターフェースを定義し、今後リーダの変更やシステム構成の変化がある場合にもこれまでのアプリケーションに影響を与えることなく変更することが可能となった。

今後は情報配信システムだけではなく、アクティブタグを用いることで実現できるアプリケーションへと対象を広げることで、API に更なる機能の拡張を実現していく。

文　献

[1] 若山史郎：Personal Server Model モデルに基づくセンサ情報管理機構の実現, 2002

[2] Yoshihiko Suko,Haruki Yokoyama,Shingo Yamada,Takashi Nagano,Yuzuru Takeuchi,Fumitoshi Kato : The demonstration of the base environment for the simulation & gaming: using real time location information and profile information, ISAGA, 2003

[3] 田辺弘実, 木原民雄：実空間メタデータ収集に基づく情報ナビゲーション, DICOMO, 2003

[4] 井上創造, 萩原大輔, 佐々木淳, 青木靖, 秋元一美, 浜崎陽一郎, 安浦寛人：RFID によるイベント支援マルチサービスシステム「RICA」の実証実験, DICOMO, 2003