RWFS: 位置情報による空間アクセス制御を行うファイルシステムの設計と実装

八木 佑侑季[1]　狐崎 直文[2]　斉藤 裕樹[3]　戸辺 義人[2]

概要：本論文では，実世界の空間にディレクトリが存在するかのように扱い，そのディレクトリに情報を置いて管理することや，同じ場所にいる者同士で情報共有をすることを可能とするファイルシステム，RWFS (Real-World File System) の設計と実装を提案する．本システムでは，管理する情報資源のアクセス権に位置情報が設定し，ユーザは自分が現在いる場所に応じたディレクトリへのアクセス権を獲得することができる．したがって，そのディレクトリや情報資源へのアクセスを試みるユーザの位置情報によって，アクセス可否をファイルシステムに判断させることが可能である．本システムで扱う場所ごとに紐づけられたディレクトリを実世界ディレクトリ (RWD: Real-World Directory) と呼ぶ．このディレクトリは，実世界の空間が有する階層構造をマッピングしたディレクトリであるため，各実世界ディレクトリ間の包含関係は，実世界の場所が有する包含関係となる．また，本研究では，実世界の場所が有する構造は単純なツリー構造ではなく，子ノードが複数の親ノードに属することが許されるような，特有の半順序構造を有していると仮定する．したがって，ある場所が複数の場所に含まれているような場合でも，場所と関連付けたディレクトリを作成することができる．本論文では，実現形態の一つとして屋内環境における RWFS を実装した結果を述べる．実際の操作時間の測定による評価，および，本システム特有のディレクトリ構造をシミュレーションによって評価した．

RWFS: Design and Implementation of File System Using Spatial Access Control By User's Location Information

Yuki Yagi[1]　Naofumi Kitsunezaki[2]　Hiroki Saito[3]　Yoshito Tobe[2]

1. はじめに

1.1 研究背景

近年，GNSS（Global Navigation Satellite System）といった位置測位技術や，スマートフォンに代表されるモバイル端末の発展が進んでいる．各人がモバイル端末を持ち，自分の現在地を取得することが容易になったことで，ユーザの位置情報を使用するアプリケーションソフトウェアが普及している．利用者の位置情報に応じて提供される位置情報サービス (LBS: Location-Based Service) はその代表例である [11][12].

1.1.1 物理空間での場所毎の情報管理

我々が生活している空間に目を向けると，場所ごとに情報が置かれ，管理されている例が多く存在する．例えば，観光地に設置される来訪者用の雑記帳は，その観光地に訪れた人が書き込み，観光者同士が共有する情報の例である [10]．ここに書かれている情報は，その場所（観光地）に実際に行きアクセスすることに有用性がある．また，壁やドアに貼られるポスターは，そのポスターの貼られている場所を通る人に対して提供される情報の例である．ポスターが提供している情報の内容は，そのポスターが貼られている場所に関連した内容であるほど，有用性を見出だすことができる．これらの例は，場所ごとに情報を置き，管理することが有効な手段であるということを示している．

1.1.2 論理空間での情報管理

我々が普段使用するコンピュータの内部で扱うような電子化された情報は，ファイルシステムによってその管理が行われている．ファイルシステムでは，ディレクトリと呼

1 青山学院大学大学院理工学研究科理工学専攻
2 青山学院大学理工学部情報テクノロジー学科
3 明治大学総合数理学部先端メディアサイエンス学科

ばれる，管理する情報をグループ化して整理する仕組みが用いられる．ディレクトリは，情報管理のし易さという観点から，階層構造を有するディレクトリが広く採用されている．階層構造によって情報をグルーピングすることは，何も構造を有していない空間で情報管理を行う場合に比べて，目的の情報へ高速にアクセスが可能となることや，利用者が情報がどこにあるのかを把握しやすくなるといった利点があることが知られている．

1.2 研究目的

観光地の雑記帳やポスターによって場所毎に管理されるような情報のように，電子化された情報も同様に実世界の場所毎に置いて管理することができれば，新しい情報管理方法を提供する位置情報サービスが提案できる．しかし，電子化された情報は本来，論理的な空間で管理されているが故に「物理空間上の場所に置く」という概念が存在しない．さらに，一般的な位置情報サービスでは，場所同士の関係や構造が考慮されていないため，従来のファイルシステムのような，ディレクトリや階層構造を意識した情報管理を行うことは難しい．そこで我々は，実世界における場所が階層構造を有しているという点に着目し，その階層構造とファイルシステムのディレクトリが有する階層構造を関連づける．本研究では，この関連付けにより，電子化された論理的な情報を，実世界のような物理空間での場所毎に置いて管理することが可能なファイルシステム，RWFS（Real-World File System） を提案する．

2. 関連研究

2.1 位置に基づく拡張現実ソフトウェア

場所と情報の結びつけに関連した技術として，位置に基づく拡張現実ソフトウェア [2] がある．これは，ユーザの位置情報に応じて実世界の場所に文字や画像を重ねて表示することができるソフトウェアである [1]．位置に基づく拡張現実ソフトウェアと本研究との違いは，次の二点が挙げられる．

2.1.1 場所の扱い方

位置に基づく拡張現実ソフトウェアでは，各場所を単体で存在する点として扱っているため，実世界の場所が持つ構造が考慮されていない．したがって，論理的な情報を物理的な空間上に置くことはできるが，その情報を管理するという観点では困難である．これに対し本研究では，実世界の場所が有する構造を考慮し，それをファイルシステムのディレクトリ構造にマッピングしている．したがって，場所に情報を置くという点と，その情報を管理しやすいという点の両方を満たしている．

2.1.2 位置の偽装の対策

位置に基づく拡張現実ソフトウェアでは，ユーザ側で位置の偽装ができてしまうといった問題がある．すなわち，システム側に虚偽の位置情報を送信することにより，本来はアクセスすることのできない情報へとアクセスすることを許してしまう．これに対し本研究では，NFC (Near Field Communication) の技術を用いることにより，ユーザの位置偽装をシステム側で防止する仕組みを有している．

2.2 実空間定義型ユビキタスシステム：空間情報サービス（USIS）

実空間定義型ユビキタスシステム：空間情報サービス（USIS: Ubiquitous Space and Inforamtion Servises）[3] は，実際にサービスや情報を提供する場所に “バルーン” と呼ばれる様々な大きさを有した仮想空間を設定し，バルーンを接触する，バルーンを掴む，バルーンを移動させるといった人間の直感的な動作でサービスを起動したり，結合したりすることが可能なサービスである．いろいろな場所に点在するバルーンは，位置情報を持ったファイルシステムとして構成／管理され，その場所の特性を生かしたビジネスやコミュニケーションを創造することが期待されるサービスである．空間情報サービス（USIS）と本研究との違いは，次の二点が挙げられる．

2.2.1 重視する側面

本研究と USIS は，場所の構造に着目した情報管理を行うという点で関連している．しかしながら，研究の目的として重視しているシステムの側面において差異がある．本研究では，場所と情報資源の保護属性が関連づけられた，ファイルシステムとしての側面を重視している．したがって，ストレージサービス [14] [15] [16] のように，システムのユーザが自由に情報を置くことが許される．これに対し USIS は，場所ごとに提供されるサービスを，場所ごとに管理するという側面を重視している．

2.2.2 場所の構造

本研究では，各場所が半順序関係を有する階層構造として扱っている．これにより，1 つの場所が複数の場所に所属するような場所を表現することができる．これに対して USIS は，場所を単純なツリー構造として扱っている．したがって，複数の場所の同時に所属するような場所を表現することは難しい．

3. RWFS の設計

RWFS は，実世界の場所と結びつけられた独自のディレクトリを有するファイルシステムである．本システムでは，ディレクトリへのアクセス権を場所毎に設定することができる．本システムはクライアントサーバモデルを採用する．ファイルシステムの本体はサーバが有しており，ユーザはクライアントアプリケーションを操作して，サーバに置かれた情報資源へとアクセスする．以上を実現するために必要とされる機能は，次の三点が必要である．(1) 実世界の空間構造を反映した構造を有するディレクトリ，

(2) ユーザの位置測位を行う機能，(3) ユーザの情報資源へのアクセス可否を判断する機能．本研究では，(1), (2), (3) の機能を持つ機構をそれぞれ，次のように名付けた．

(1) 実世界ディレクトリ
(2) Locator
(3) 保護機構

3.1 実世界ディレクトリ

実世界ディレクトリは，実世界の場所が有する階層構造をディレクトリ構造へとマッピングさせたディレクトリである．実世界ディレクトリのアクセス権には，対応する実世界の位置情報が設定される．また，その実世界ディレクトリ上で管理される情報資源には，その実世界ディレクトリと同じアクセス権が設定される．したがって，RWFS のユーザは，実世界ディレクトリと関連づけられた場所に行くことによって，実世界ディレクトリにアクセスすることができる．

3.1.1 用語の定義

実世界ディレクトリの構造を説明する際に必要となる用語についての定義を行う．

場所の包含関係

場所の集合 $P = \{p_1, p_2, \cdots, p_n\}$ を考える．場所 p_q と $p_r (p_q \neq p_r)$ を考えた際に，p_q が p_r に含まれることを二項関係 $\in$ を定義し， $p_q \in p_r$ と表すものとする．

半順序集合

順序集合のある 2 つの要素間の二項関係において，「どちらが大きいか（小さいか）」などといった，比較不可能であるような関係がある要素を含む集合を半順序集合という．要素間の二項関係を表す記号を $\leq$ とすると，半順序関係は次のように定義される．

(1) $\forall a, a \leq a$
(2) $a \leq b \wedge b \leq a \Rightarrow a = b$
(3) $a \leq b \wedge b \leq c \Rightarrow a \leq c$

場所を要素として有する集合 P を考えた場合，各要素には「含む」や「所属する」といった包含関係 $\leq$ が存在する．関係 $\leq$ は，反射律，反対称律，推移律を満たすため順序関係であるといえるが，完全律を満たさないため全順序集合ではない．したがって， $(P, \leq)$ は半順序集合であるといえる．

鎖と極大鎖

場所の集合 P において，$p_i \in p_{i+1} \in \cdots \in p_j$ を満たすとき，$\{p_i, p_{i+1}, \cdots, p_j\}$ を P の鎖という．また，$x \in y (x \neq y)$ において，$z \neq x, y$ かつ $x \in z \in y$ となるような z が存在しないとき，x, y はカバー関係にあると呼び，「y は x をカバーしている」という．
場所の集合 P の鎖 $\{p_i, p_{i+1}, \cdots, p_j\}$ において，各 p_i が p_{i-1} をカバーしているとき，この鎖は飽和しているといい，飽和している鎖を極大鎖と呼ぶ．

ハッセ図

本研究で扱うディレクトリ構造は半順序関係を有するが，これを視覚的に表現する方法としてハッセ図がある．ハッセ図では，半順序関係を有する集合の個々の元を頂点とし，カバー関係にある要素間を線で結んで表現する．この順序関係では，ある要素よりも上位にあるものを，その要素よりも上側に置いて表現する．

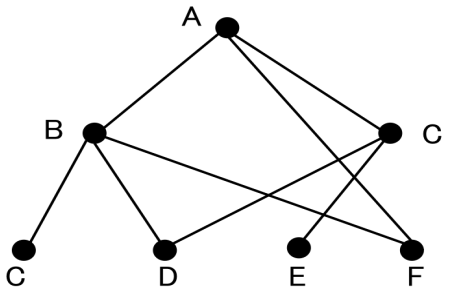

図 1 ハッセ図の例

図 1 のハッセ図の例では，半順序関係を有する $\{A, B, C, D, E, F\}$ という集合の要素に対して，その関係を表現している．ここでの要素間の二項関係を表す記号を $\leq$ とすると，ハッセ図から，$A \leq B, A \leq C, B \leq C, B \leq D, B \leq F, C \leq D, C \leq E$ であることがわかる．

3.1.2 実世界ディレクトリの構造

実世界における場所を考えたとき，これを必ずしもツリー構造で表現できるとは限らない．その理由として，順序関係が定義できない複数の場所に同時に所属するような場所が存在するためである．例えば，図 2 に示すような場所の場合，$Stairs$ は $Floor1$ と $Floor2$ という，順序が定義されていない 2 つの場所に同時に所属している．このような場所が存在する場合，ツリー構造で表現することができない．

本研究では，実世界における場所を要素として有する集合を考えた場合に，その集合は複数の親を有する場合が存在する半順序集合であるということに着目して，場所の構造化を行う．場所の構造化を行った後，それをファイルシステムのディレクトリ構造へとマッピングする．
まず，実世界における場所が有する構造は，場所を頂点とするハッセ図で表現することができる．このような手順で場所を表現したハッセ図の各頂点と一対一で対応する位置

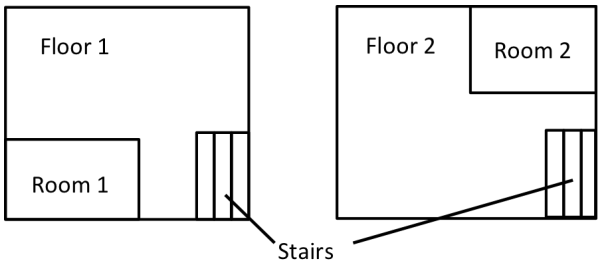

図 2　実世界における場所の例

にディレクトリを作成し，そのハッセ図と同様の階層包含関係を有するように作成したディレクトリ構造が実世界ディレクトリと定義する．実世界ディレクトリとして，図 1 に示す場所における例を述べる．

$$\text{Places} = \{\text{Floor 1, Room 1, Stairs, Floor 2, Room 2}\} \quad (1)$$

図 2 において, 場所を要素として有する集合 $Places$ は, 集合 (1) のように表すことができる．この例において，$Room1$ は $Floor1$ に，$Room2$ は $Floor2$ に属しており，$Stairs$ は $Floor1$ にも $Floor2$ にも属している空間であると考えることができる．したがって，図 1 に示した場所をハッセ図で表現すると，図 3 の左図のように表現することができる．このハッセ図から作成した実世界ディレクトリを，図 3 の右図に示す．

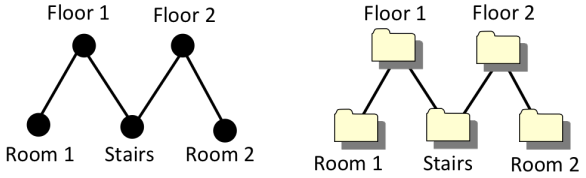

図 3　ハッセ図と実世界ディレクトリ

3.1.3　実世界ディレクトリへのアクセス

場所の集合における要素 p_a, p_b, p_c があり，$p_b \leq p_a$ かつ $p_c \leq p_b$ である場合，推移律より $p_c \leq p_a$ を満たす．これは，ある人物が p_c という場所にいることは，同時にその人物が p_b という場所と，p_a という場所にいることを意味する．例えば，渋谷区にいることは東京都にいることを意味するし，同時に日本にいることも意味する．実世界ディレクトリでは，この性質をアクセス制御に用いる．

本システムでは，ユーザがアクセス可能な実世界ディレクトリは，次の 2 つの条件を満たすものであると定義する．

- ユーザがいる場所に対応した実世界ディレクトリ d
- d から，上へと伸びる極大鎖上のすべての頂点に対応する実世界ディレクトリ

例として，3.1.2 節 で挙げた 図 3 における実世界ディレクトリのアクセスの可否を考えると，ユーザが現在いる場所に対するアクセス可能な実世界ディレクトリの関係は，表 1 のようになる．

表 1　ユーザの場所とアクセス可能なディレクトリ

ユーザの場所	アクセス可能なディレクトリ
$Floor1$	$Floor1$
$Room1$	$Room1$, $Floor1$
$Stairs$	$Floor1$, $Floor2$
$Floor2$	$Floor2$
$Room2$	$Room2$, $Floor2$

3.2　Locator

Locator は，ユーザの位置推定機能を有する機構である．Locator には任意の位置推定技術を実装することができ，RWFS のユーザがどこにいるか，システム側が把握できるようにする．さらに Locator は，近距離無線通信を用いることによって，ユーザの位置の偽装をシステム側で防ぐ仕組みを有する．

3.2.1　位置推定技術

本システムで扱う場所は，屋内や屋外といった場所の種類や認識の精度問わず，Locator に実装する位置推定技術により柔軟に変更することができる．その際に決めておくべき事として，次の 2 点がある．

(1) クライアントが送信する位置推定用の情報
(2) システムで扱う場所

Locator では， RWFS で扱う場所の種類に限らず，この 2 点を対応付ける情報が必要である．例えば，電波強度を用いた屋内位置測位技術を使用する場合，クライアントが送信する位置推定用の情報として,「クライアントが受信した電波強度のセット」を扱う．GPS を用いて屋外での場所を扱う場合は,「 GPS で受信した緯度経度情報」といった情報を扱う．これに対しシステムで扱う場所は， RWFS 内で実際に場所として見なす場所，すなわち，実世界ディレクトリと一対一で対応付けられる場所を指す．Locator では，これら 2 つを紐付ける情報さえあれば，多くの位置推定技術を扱うことができる．

3.2.2　位置偽装の防止

ユーザの位置情報を使用するモバイルアプリケーションは，使用する位置推定技術によって位置情報の偽装ができてしまうという問題がある．本システムも，ユーザの位置推定に必要な情報を，クライアントアプリケーションを通してサーバに通知する設計である．したがって，ユーザが虚偽の位置情報をシステムに送信することで，本来はアクセスを許可されていない実世界ディレクトリへアクセスを許してしまう恐れがある．これを防ぐために，本システムは，NFC（Near Field Communication）を用いて，ユーザの位置偽装をシステム側で防止する仕組みを有する．NFC

とは，13.56MHz の周波数帯の電磁波を利用した近距離無線通信技術のことである [13]．NFC では，NFC 通信規格に対応した IC チップを搭載している機器やカードを，数 cm の距離でかざすことによって，データの読み込みや書き込みといった通信を行うことができる． 表 2 に NFC とその他の近距離無線通信規格についての比較を示す．

表 2　NFC とその他の技術の比較

	NFC	RFID	IrDa	Bluetooth
セットアップ	∼0.1ms	∼0.1ms	∼0.5s	∼6sec
通信距離	∼10cm	∼3m	∼5m	∼30m

NFC は，通信をする際に要するセットアップの時間が少ないことや，他の近距離無線通信規格と比べて，通信距離が短いことが特徴であるといえる．逆説的に述べると，NFC を用いて長距離の通信を行うことはできない．この性質を利用し，本システムでは，ユーザが位置情報を偽装することを防止するための仕組みとして，NFC を用いた追加認証を行わせることが可能である．

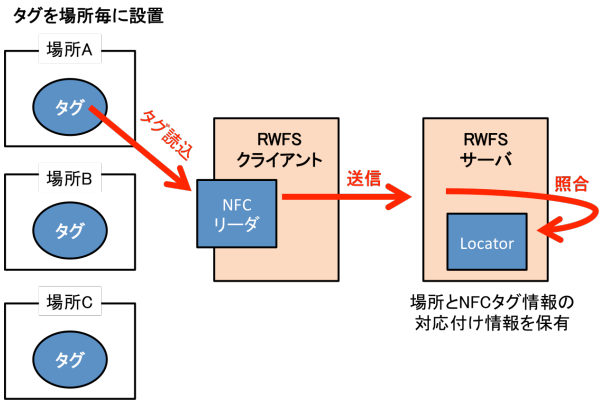

図 4　NFC による位置偽装防止

図 4 に示すように，場所毎に予めタグを埋め込んでおく．追加認証を行う場合，ユーザは RWFS クライアントと接続した NFC リーダによってタグの情報を読み取る．この情報をそのままサーバへ送信させることで，サーバ側で認証を行う．この作業を行うことによってユーザは，その場所にいることが保証される．

3.3　保護機構

保護機構は，RWFS が管理する情報資源のメタデータや，情報資源ごとに割り当てられたアクセス権を管理する機構である．Locator で測位したユーザの位置情報を基に，ユーザにファイルシステム操作の可否を決定させる．

3.3.1　保護ドメイン

RWFS サーバが保有する情報資源を保護するために，本システム独自の保護ドメインの概念を定義する．保護ドメインとは，情報資源とそれに対する権限の組をひとまとまりで表すものである． RWFS では，保護ドメインを実世界の各場所ごとに保有する．したがって，RWFS のユーザが実世界の場所を移動する度に保護ドメインが切り替わる．

3.3.2　ケーパビリティ

ケーパビリティとは保護ドメインに含まれる保護対象の情報資源と，それに対するアクセス権限を関連づける概念である．ケーパビリティは，保護ドメインごとに管理されるケーパビリティリストに格納されており，保護機構は情報資源に対応するケーパビリティを使用してアクセスの可否を判断する．

3.4　RWFS 全体のやりとり

図 5 に RWFS 全体のやりとりを示す．

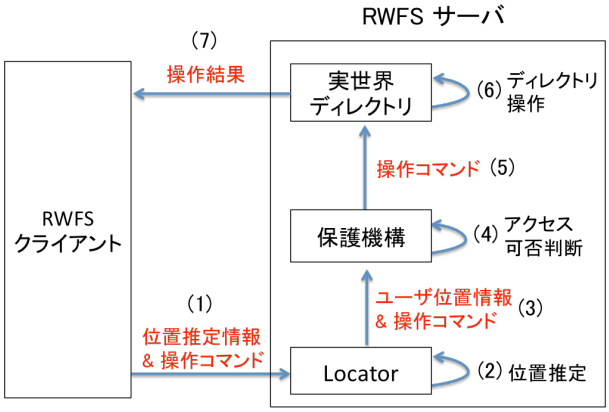

図 5　RWFS 全体のやりとり

図 5 に示した RWFS 全体の処理を説明する．

(1) クライアントは，ファイルシステムを操作するためのコマンドと，自分の位置をシステムに知らせるための位置推定情報をサーバへ送信する．
(2) サーバの Locator が情報を受け取り，位置推定情報を使用して，ユーザの位置推定を行う．この段階で，ユーザの位置情報が確定する．
(3) ユーザの位置情報とファイルシステムの操作コマンドを，保護属性へと引き渡す．
(4) 保護属性では，ユーザの位置情報に応じて，ファイルシステムの操作をするためのアクセス権があるか否かを判断する．
(5) アクセス権がある場合，操作コマンドが実世界ディレクトリへ送る．
(6) 実世界ディレクトリに対して操作コマンドを実行し，実際のファイルシステムの操作を行う．
(7) 操作結果をクライアントへ送り返す．

4.　実装

本研究の設計に対する実装の一例として，図 6 に示すようなモジュールで実装を行った．

図 6 に対する各モジュールの説明は，次の表 3 と 表 4 のとおりである．

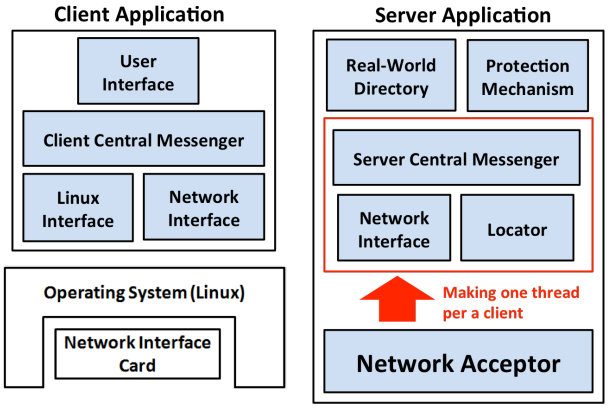

図 6 実装の構成

表 3 実装したクライアントサイドの各モジュールの説明

モジュール名	説明
User Interface	ユーザがクライアントアプリケーションを操作するためのインタフェース
Client Central Messenger	クライアントの各モジュールに対し制御命令を出す中心的モジュール
Linux Interface	Linux に命令を送信し，NIC の情報を得るためのインタフェース
Network Interface	サーバと通信を行うためのネットワーク機能を有するモジュール

表 4 実装したサーバサイドの各モジュールの説明

モジュール名	説明
Real-World Directory	実世界ディレクトリの実体を保持するモジュール
Protection Mechanism	ユーザの情報資源へのアクセス可否を行う保護機構モジュール
Server Central Messenger	サーバの各モジュールに対し制御命令を出す中心的モジュール
Network Interface	クライアントと通信を行うためのネットワーク機能を有するモジュール
Locator	Locator モジュール
Network Acceptor	マルチクライアントを実現するために，サーバのスレッド化を行うモジュール

4.1 RWFS サーバの実装

Locator には，ユーザの位置を測定する技術として，Wi-Fi 電波強度を用いた Fingerprint 法による屋内位置測位技術 [[4], [5], [6], [7]] を実装した．RWFS サーバと RWFS クライアントは TCP ソケット通信によってデータを送受信させる．Locator で対応付けるべき，クライアントが送信する位置推定用の情報，および，システムで扱う場所は，次のようなフォーマットで定義を行った．

```
<fingerprint>
    <RSSIs>
        <RSSI RSSIvalue="-40.10" apName="AP1"/>
        <RSSI RSSIvalue="0.0" apName="AP2"/>
        <RSSI RSSIvalue="-44.83" apName="AP3"/>
        <RSSI RSSIvalue="-42.50" apName="AP4"/>
        <RSSI RSSIvalue="0.0" apName="AP5"/>
        <RSSI RSSIvalue="0.0" apName="AP6"/>
        <RSSI RSSIvalue="-39.33" apName="AP7"/>
    </RSSIs>
    <place name="Tobe-Labo" id="15810087"/>
</fingerprint>
```

この例では，予め屋内環境に Wi-Fi のアクセスポイントを設置し，場所として登録する地点で，それらのアクセスポイントから受信した電波強度を記録したものである．複数の電波強度と，1 つの場所の名前の対応付けを示している．

4.2 RWFS クライアントの実装

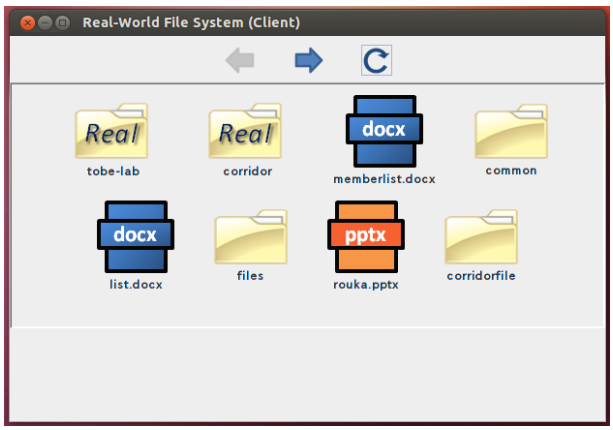

図 7 RWFS クライアントの操作画面

図 7 は RWFS クライアントを操作している画面である．RWFS のユーザは，ここで示されているクライアントアプリケーションの画面を通して，サーバにあるファイルシステムの実体を操作する．

図 8 RWFS と NFC

図 8 は RWFS クライアントと NFC リーダを実装したものである． 実世界の場所に固有の NFC タグを埋め込み，RWFS クライアントから NFC リーダを通して NFC

タグに書かれた情報を読み取り，システム側でユーザの位置偽装を防ぐことが可能となる．

5. 評価

RWFS はクライアント・サーバモデルを採用したシステムであるため，ユーザはネットワークを通してファイルシステムを操作する．したがって，ファイルシステムの操作に伴うオーバヘッドが懸念される．そこで，RWFS の稼働に必要な動作や，ファイルシステムの基本的な操作を行うのにかかる時間を測定した．また，本システムのディレクトリ構造は半順序構造を有する．そこで，一般的なファイルシステムで使用されているツリー構造のディレクトリと比べ，本システムで扱う半順序構造のディレクトリの場合に，ノードからノードへのアクセスに要するホップ数の違いをシミュレーションによって計算した．

5.1 基本操作性能の評価

ファイルシステムを操作する際のターンアラウンドタイムを測定する．行う操作内容は以下のとおりである．

(1) ディレクトリ変更
(2) ディレクトリ作成
(3) ファイルアップロード

次に実験結果を示す．

表 5 基本操作に要するターンアラウンドタイム

操作	平均 [ms]	標準偏差 [ms]
ディレクトリ変更	326	88
ディレクトリ作成	338	68

表 6 ファイルのアップロードに要する時間

データサイズ	平均 [ms]	標準偏差 [ms]
1 KB	310	81
10 KB	169	51
100 KB	372	166
1000 KB	1942	591

表 5 はファイルシステムの基本操作に要するターンアラウンドタイムを計測した結果である．ターンアラウンドタイムとは，クライアント側で操作指示を出した瞬間から，クライアント画面に操作結果の表示が完了するまでに要する時間のことである．また， 表 6 は，サーバにファイルをアップロードする際に要する時間を計測した結果である．以上の評価実験により，ファイルシステムの操作に必要な基本的な操作を少ないオーバヘッドで実行できることが確認できる．

5.2 ディレクトリ構造の評価

RWFS で扱う特殊なディレクトリ構造をシミュレーションによって評価した．シミュレーションの内容は次のとおりである．

(1) ツリー構造のディレクトリをランダム作成
(2) そのディレクトリに対し，ランダムに決めた 2 ノード間のホップ数を計算
(3) (2) を 1000 回繰り返し行い，平均ホップ数を算出
(4) (1) で作成したツリーを変形し，半順序構造にする
(5) そのディレクトリに対し，ランダムに決めた 2 ノード間のホップ数を計算
(6) (5) を 1000 回繰り返し行い，平均ホップ数を算出

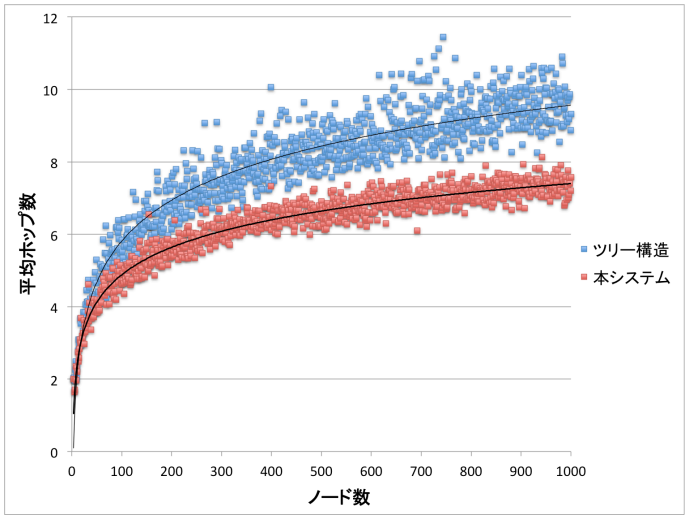

図 9 ツリー構造と半順序構造の平均ホップ数の比較

図 9 に示したものは，ツリー状のディレクトリ構造の場合と，本システムで扱うディレクトリ構造を扱った場合の散布図と，それらの近似曲線である．どちらも指数関数近似が可能であるが，このシミュレーション結果では，一般的なファイルシステムに使用されているツリー構造を有するディレクトリよりも，本システムで扱うディレクトリ構造の方が，ランダムに決めた 2 ノード間を移動する際の平均ホップ数が少なくなることが示された．

6. 展望

6.1 位置推定技術の変更

本論文では，Locator に Wi-Fi 電波強度の Fingerprint 法を用いた屋内位置測位を実装したが，Locator に実装する位置推定技術を変更してシステムの評価をする．その一例として，GPS[17] やその他の屋外位置推定技術 [8]，無線 LAN を用いた位置推定技術 [9] を使用し，システムで扱う場所の拡張を行う．これにより，屋外の場所で情報管理

や，屋外の同じ場所にいる者同士で情報を共有できる，情報サービスを提案することができる．

6.2 位置偽装への対応

この問題に対する解決策は多くの研究がなされている [18], [19], [20]．本論文では，NFC を用いた位置偽装への対策を提案したが，今後は，既存の研究を基に位置偽装に対する新たな対策を提案する．

7. むすび

本論文では，実世界の場所ごとに情報を結びつけて管理することができるファイルシステム，RWFS について述べた．RWFS を用いることによって，コンピュータ内部で扱われる論理的な情報を，実世界という物理空間と結びつける新しい情報管理が可能となった．また，その実装を行い，少ないオーバヘッドでシステムの操作が可能であることを示した．加えて，シミュレーションによって，本システム独自のディレクトリ構造が一般的なファイルシステムで使用されるツリー状のディレクトリ構造よりも，少ないホップ数でアクセス可能であるという知見が得られた．今後は，新しい情報管理の方法として普及することが期待される．

参考文献

[1] Paucher, R., Turk, M.: Location-based augmented reality on mobile phones, CVPRW, 13-18 June (2010).

[2] Geiger, P., Schickler, M., Pryss, R., Schobel, J., and Reichert, M.: Location-based Mobile Augmented Reality Applications: Challenges, Examples, Lessons Learned, (2014).

[3] 森信一郎, 畠添菜美, 塩内正利, 原政博, 藤野信次: 実空間定義型ユビキタスシステム: 空間情報サービス (USIS)(情報配信サービス), 情報処理学会研究報告, MBL,[モバイルコンピューティングとユビキタス通信研究会研究報告] 2005.113, pp. 201-206, (2005)

[4] Kaemarungsi, K., and Krishnamurthy, P.: Modeling of Indoor Positioning Systems Based on Location Fingerprinting, in Proc. IEEE INFOCOM, May (2004).

[5] Wang, H., Sen, S. Elgohary, A., M. Farid, M. Youssef, and R. Choudhury.: No need to war-drive: Unsupervised indoor localization, In Proceedings of ACM MobiSys, (2012).

[6] Azizyan, M., et al.: SurroundSense: Mobile phone localization via ambience fingerprinting, In MOBICOM, (2009).

[7] Martin, E., Vinyals, O., Friedland, G. and Bajcsy, R.: Precise indoor localization using smart phones, Proceedings of the international conference on Multimedia, ACM, (2010).

[8] 暦本純一, 塩野崎敦, 末吉隆彦, 味八木崇.: PlaceEngine: 実世界集合知に基づく WiFi 位置情報基盤, インターネットコンファレンス, pp. 95-104, (2006).

[9] LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, J., Borriello, G. and Schilit, B.: Place lab: Device positioning using radio beacons in the wild, In Pervasive computing (PERVASIVE 2005), pp. 116-133, (2005).

[10] 岡本健.: アニメ聖地における巡礼者の動向把握方法の検討: 聖地巡礼ノート分析の有効性と課題について, 観光創造研究= Advanced Tourism Studies 2, pp. 1-13, (2008).

[11] Chen, P. T. and Lin, Y. S.: An Analysis on Mobile Location-based Services, Commun, ACM, Vol. 46., No. 12, pp. 61-65, (2003).

[12] Rao, B. and Minakakis, L.: Evolution of mobile location-based services, Communications of the ACM 46.12, pp. p61-65, (2003).

[13] Yaqub-Undergraduate, M. U. and Shaikh-Undergraduate, U. A.: Near Field Communication, (2012).

[14] Dillon, T., Wu, C. and Chang, E.: Cloud computing: issues and challenges, Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on. Ieee, (2010).

[15] SO, K.: Cloud computing security issues and challenges, International Journal of Computer Networks (2011).

[16] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A. and Zaharia, M.: A view of cloud computing, Communications of the ACM 53.4, pp. 50-58, (2010).

[17] Kaplan, E. D. and Hegarty, C. J.: Understanding GPS: Principles and Applications, Artech House, Boston, MA, (1996).

[18] Zhu, Z. and Cao, G.: Applaus: A privacy-preserving location proof updating system for location-based services, INFOCOM, 2011 Proceedings IEEE. IEEE, (2011).

[19] Capkun, S. and Hubaux, J. P.: Secure positioning of wireless devices with application to sensor networks, INFOCOM 2005, 24th Annual Joint Conference of the IEEE Computer and Communications Societies, Proceedings IEEE, Vol. 3, IEEE, (2005).

[20] Saroiu, S. and Wolman, A.: Enabling new mobile applications with location proofs, In ACM HotMobile, (2009).