

P2Pエージェントプラットフォーム PIAX の理念と実装

吉田 幹 (株)ビービーアール,
大阪大学(招聘研究員)



PIAXの設計コンセプト

共通プラットフォームのニーズ

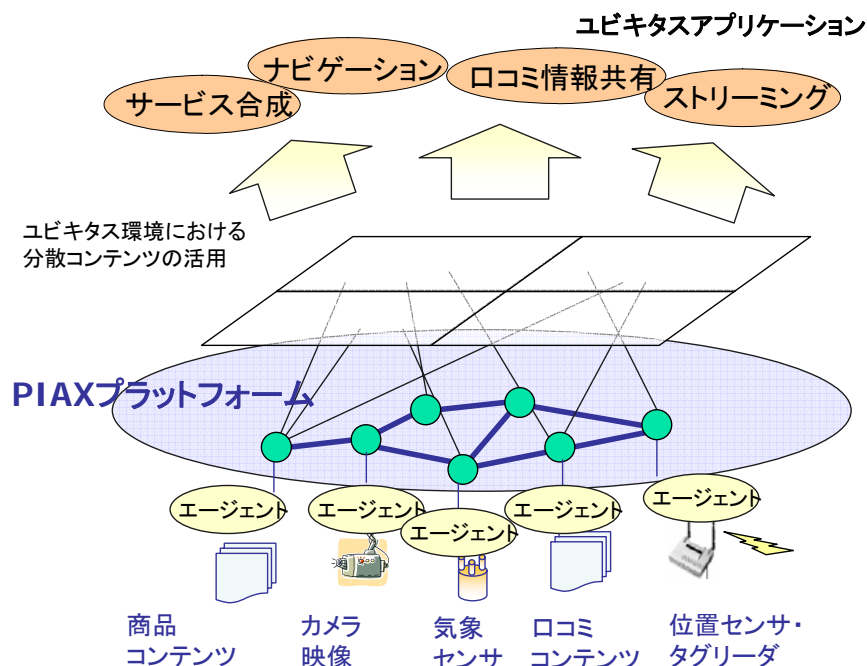
- ユビキタスネットワーク
 - PCに加え、携帯電話、PDA、家電機器、自動車、センサ、ロボット、ウェアラブルなど、多種多様のデバイスがネットワークに接続
 - 3C(Computing, Connectivity, Contents) everywhere
- 課題
 - 大量のオブジェクトから必要なものを瞬時に発見する
 - 発見したオブジェクトを使って高度なサービスを構成する

ユーザの位置や、情報間の関係に基づくオブジェクトの発見と連携をスケーラブルに実現するための共通プラットフォームが必要

3

ユビキタスサービスの実現イメージ

- PIAXは共通プラットフォームとして機能



4

近傍性、近接性と相互作用

＜ユビキタスコンピューティングの処理の流れ＞

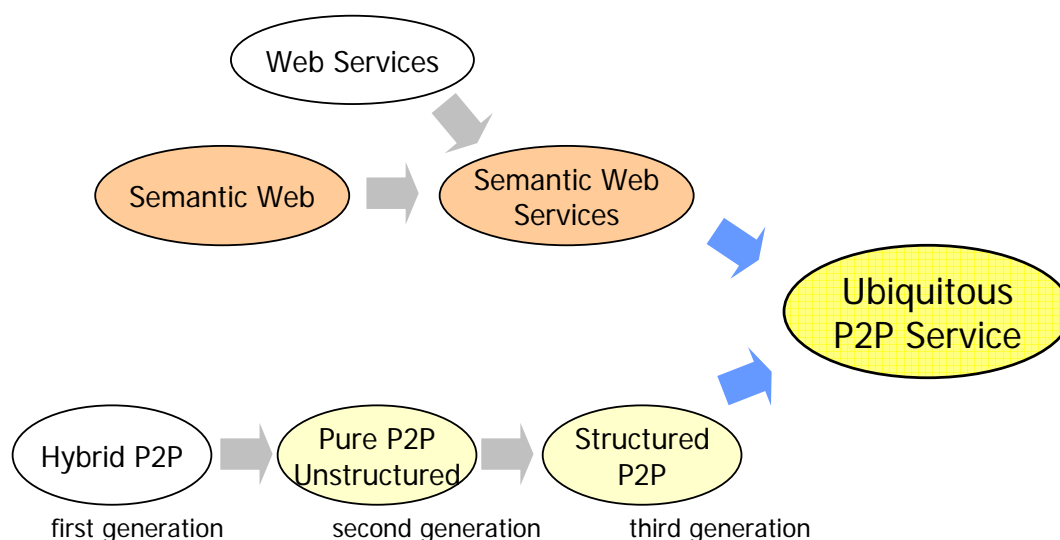
1. ユーザ位置を同定
2. 近傍、近接オブジェクトの探索
 - 近傍に位置するオブジェクト
⇒ 物理空間における**近傍性**
 - 仮想空間上に蓄積された関連性のあるオブジェクト
⇒ 意味空間における**近接性**
3. 探索されたオブジェクト(複数のサービス主体と能動的オブジェクト)の相互連携によりサービスを実現

「ユビキタスサービスは、ユーザ(群)とその近傍、近接にあるオブジェクト間における相互作用により実現される」

5

ユビキタスP2Pサービス

- セマンティックWebとP2Pシステムの進化融合形
 - セマンティックWebサービスから見れば、**P2P拡張**
 - P2Pシステムから見れば、**第4世代P2P**として知識高度化



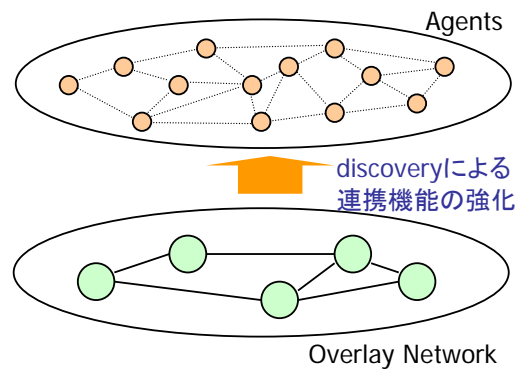
6

P2Pエージェントプラットフォーム PIAX

- ユビキタスP2Pサービスの1つの実現形
- P2P Interactive Agent eXtentions の略
(エージェントの相互作用を表現)

■ P2Pエージェント

- P2Pネットワーク上に分散する相互連携性と自律性を持ったオブジェクト
- サービス主体と能動的オブジェクトを含めたオブジェクトを表現
- オーバレイネットワークの持つ強力な資源探索機能を活用することで、“what to find” をスケーラブルに実現



7

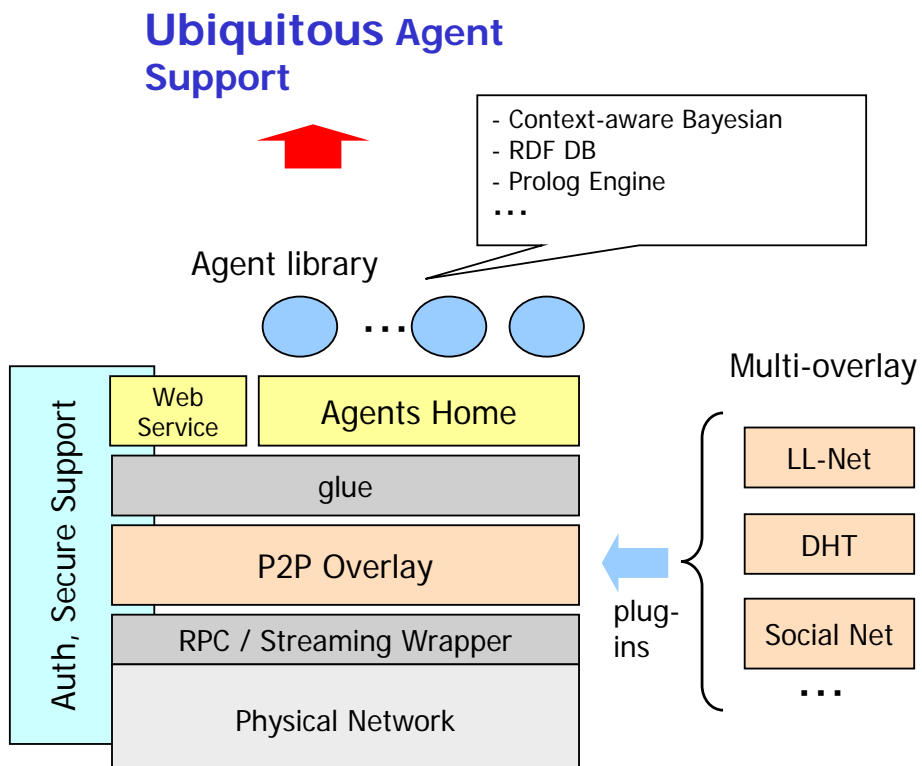
P2Pエージェントのモビリティ

- PIAXでは、エージェントに対し弱モビリティ(スタックを含むcontextの移動はしない)をサポート
- P2Pシステムにおける利点
 - 帯域の効率的利用
 - 通信頻度が高い相手の近傍にエージェントを移動させることで、ネットワーク全体の帯域消費を抑える
 - 負荷の平準化
 - 負荷の高いエージェントを複製し、リクエスト元の近傍に移動させることで、全体負荷を平準化させる
 - サービスの持続性
 - ユーザが利用するアクセスピアの変化に対し、サービスを追従させる
 - P2Pアプリケーションの耐性
 - P2Pネットワークの構成変化(churn)に対するアプリケーション側の耐性を高める

8

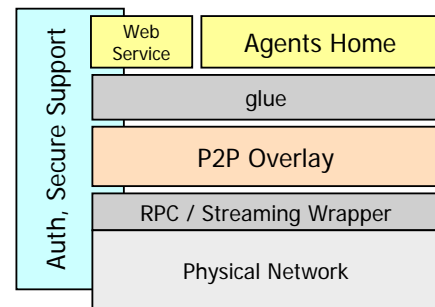
PIAXの設計

PIAXのアーキテクチャ (1)



PIAXのアーキテクチャ (2)

- RPC / Streaming Wrapper
 - Socket通信の隠蔽、NAT越えのサポート
 - 上位レイヤに対し、ピアIDを使った高レベルの通信手段を提供
- P2P Overlay
 - 複数のオーバレイ機能を管理
 - DHT, LL-Net, ソーシャルネット等
- Agents Home
 - P2Pエージェントの稼働環境(API)を提供
- Web Service
 - 外部システムとの通信手段を提供
- Auth, Secure Support
 - ピアおよびエージェントの認証等



11

マルチオーバレイ

- 共通的なプラットフォームとして機能するためには、アプリケーションからの様々な探索要求に応える必要がある
- ↓
- 複数のオーバレイネットワークを管理
 - オーバレイネットワークのプラグイン化

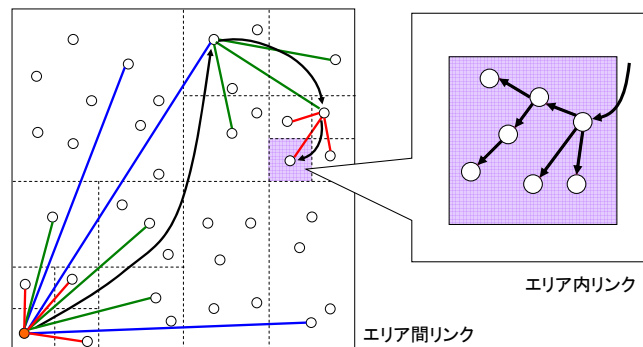
クエリパターン	オーバレイネットワーク
完全一致探索	DHT, (Skip Graph)
部分一致探索	DHT + α
範囲探索(1またはN次元)	Skip Graph, LL-Net
連想探索(SNSなど)	Unstructured
意味探索	DHT + α

※複合的なクエリパターンを処理する機構(インターリーブ)については今後の課題

12

現サポートのオーバーレイネットワーク (1)

- LL-Net (Location-based Logical P2P Network)
 - 地理的探索のためのオーバーレイネットワーク
特に、近傍オブジェクトの探索に効力を発揮
 - ピアの位置(緯度、経度)に基づいてオーバーレイネットワークを構築・維持
 - 対象世界(地球全体)を矩形のエリアに分割し、階層的に管理
 - モバイル端末の接続・切断・移動に応じて、オーバーレイネットワークを動的に構成



13

現サポートのオーバーレイネットワーク (2)

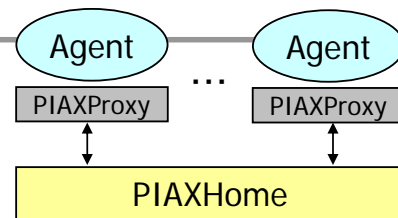
- DHT
 - (key, value) のペアをkeyのハッシュ値に従い分散管理
- ソーシャルネット(ヒューマンネットとも呼んでいた)
 - SNSのバックボーンとなるオーバーレイネットワーク
 - ユビキタスでは、ロコミ情報収集に活用
 - ⇒ 人のセンサ化
 - 人の関係を単方向グラフで表したリンク構造を元に形成
 - リンクには、複数の属性とそれに対する重み(信頼度)を割り付ける
 - 以下のクエリにより、必要な情報が得られる
 - query(最大返答数, 最大hop数, 属性, 重み)

14

P2PエージェントAPI (1)

P2Pエージェントの状態遷移

- 生成
 - `PIAXProxy PIAXHome# createAgent(String clazz, String kind, String name)`
※ kindは、同報型、発見型メソッド呼び出しで使う
- 消滅
 - `void PIAXProxy#disposeAgent()`
- 複製
 - `PIAXProxy PIAXProxy#cloneAgent()`
- 永続化
 - `void PIAXProxy#deactivateAgent(int duration)`
- 永続状態からの復帰
 - `void PIAXProxy#activateAgent()`
- 他ピアへの移動
 - `void PIAXProxy#dispatchAgent(String peerId)`



15

P2PエージェントAPI (2)

同期型、非同期型メソッド呼び出し

- ローカルピア同期型
 - `Object call(PIAXProxy proxy, String method, Object[] args)`
- リモートピア同期型
 - `Object call(String peerId, String agentId, String method, Object[] args)`
- ローカルピア非同期型
 - `FutureReturn callAsync(PIAXProxy proxy, String method, Object[] args)`
※ FutureReturn ... 返り値を非同期的に取り出すためのクラス
- リモートピア非同期型
 - `FutureReturn callAsync(String peerId, String agentId, String method, Object[] args)`

16

P2PエージェントAPI (3)

同報型、発見型メソッド呼び出し

- ローカルピア同報型
 - ReturnSet callMulti(String kind, String method, Object[] args)
 - 指定したkindを持つP2Pエージェントにのみ同報される(以降同じ)
 - ※ ReturnSet ... 複数の返り値を非同期的に取り出すためのクラス
- リモートピア同報型
 - ReturnSet callMulti(String peerId, String kind, String method, Object[] args)
- **発見型**
 - P2Pエージェントの探索とリクエストを同時に行う
 - ReturnSet callMulti(MatchingSpec mspec, String kind, String method, Object[] args)
 - LL-Netの場合の呼び出し例:

```
rset = callMulti(new Rectangle(135.0, 35.0, 1.0, 1.0),  
                "item",  
                "getScore", new Object[] {myProfile});
```

17



PIAXの実装 – Skip Graphについて

Skip Graph

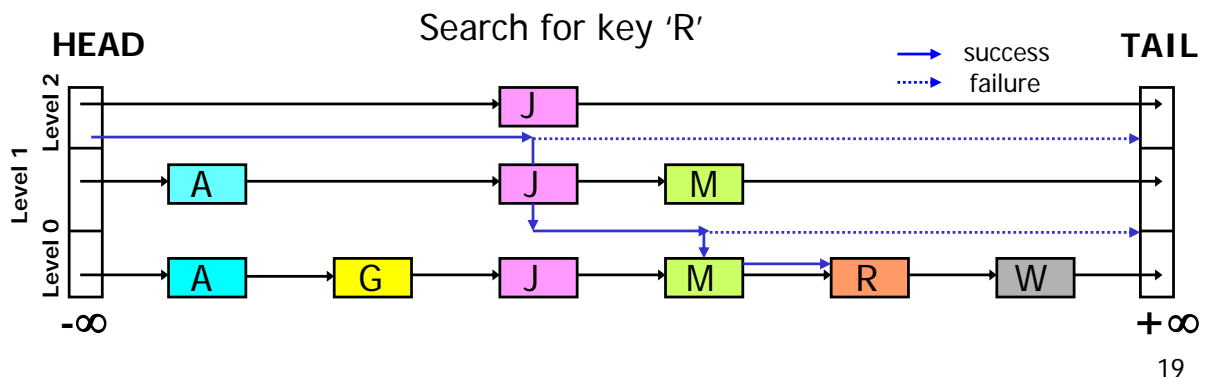
- 論文

J. Aspnes and G. Shah. Skip Graphs. *In Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, Jan. 2003.

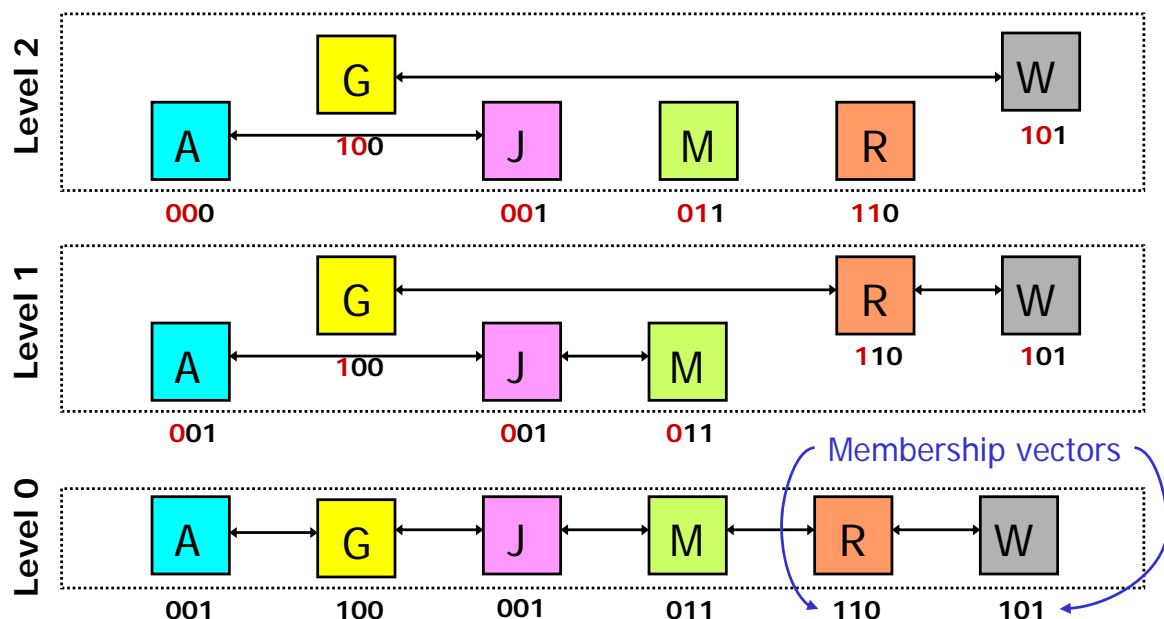
同様の研究に、SkipNetがある

- 機構

- ベースはSkip List (W. Pugh1990 バランス木実現の1手法)
- DHTで格納したいデータ位置を制御できない問題に対処(locality制御)
- ピアには、IDとkeyが振られ、どちらをキーとした検索もlog Nで実現できる
- ここで、IDはユニークで均等に分散していることが前提
- keyは順序を持った識別子で片寄りがあってもよい。範囲検索が可能



Skip Graphの基本構造



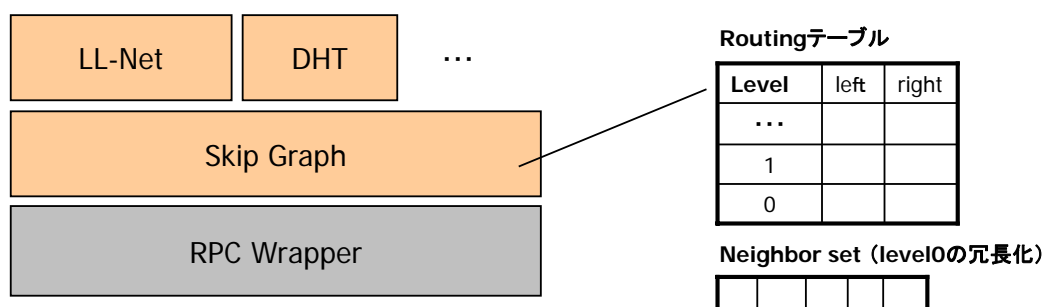
Skip Graphの利点

- IDの検索
 - IDと(IP, port)の動的変更に耐えるシステムを実現できる
 - 加えて、DHTも実現可能
- keyの範囲検索
 - L-ID(ピアの位置を表す64ビットデータ)の概念は、2次元情報(緯度経度)の1次元マッピングに相当
 - 特定エリアを指定する探索は、ある範囲に含まれるL-IDを持つピアの探索に相当
 - LL-Netのクエリーは、Skip Graphのkey範囲指定探索を使って実現できる (正確には、Skip Graphには範囲指定探索の機能について言及していない。今回、こちらで拡張した分)
- 耐障害性
 - レベル0のリングが切れない限り、探索を継続できる
→ レベル0のリングについてのみ冗長構成をとればよい
 - DHTと違い、資源を使う側の近傍に配置できるため、バースト的なピアダウンによるネットワーク分断に強い
 - メモリー効率と強い耐障害性の両方を実現

21

レイヤー構成図

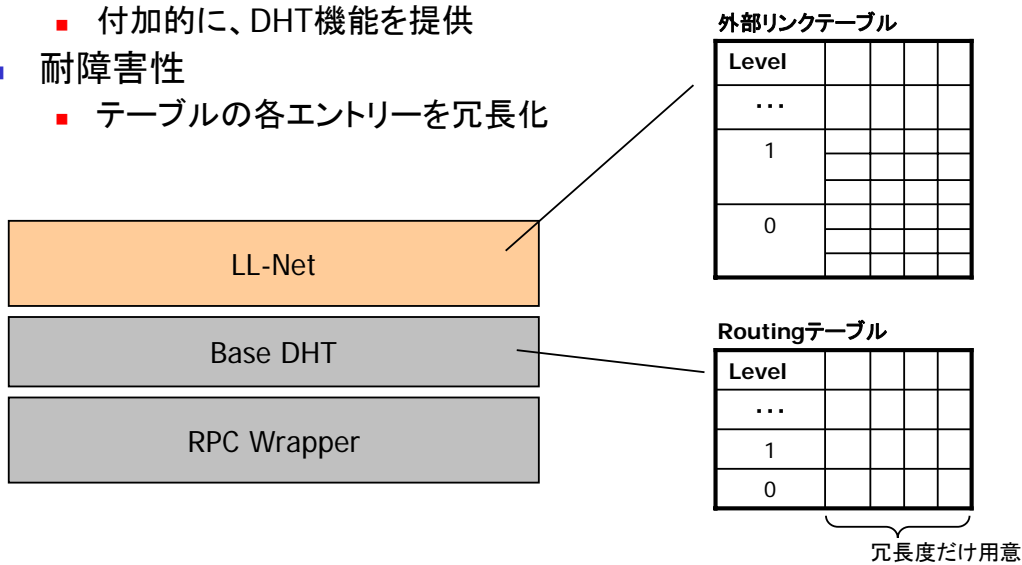
- Skip Graph
 - LL-NetとDHT機能をサポート
- Ringテーブル
 - Skip Graphの基本テーブル
 - 両方向リンク。テーブルの高さはピアIDのビット長
 - 隣接セットは、レベル0エントリーの冗長リスト。16が推奨値
- LL-Net, DHTのオーバーレイ
 - Skip Graphの機能を使って実装
 - 他にもキーワードベクトルを使ったコンテンツマッピング用のオーバーレイなど対応可能



22

レイヤー構成図(改良前)

- RPC層
 - リレーピアによる中継機能を仮想化し、上位レイヤーにネットワークの物理構造を隠蔽
 - (peerId, ip addr, port) を基本情報として用いる
- ベースDHT
 - ピアIDと(IP, port)の動的な対応付け
 - 付加的に、DHT機能を提供
- 耐障害性
 - テーブルの各エントリーを冗長化



23

参考:P2P技術

- P2P技術 分類の視点
 - Hybrid or Pure
 - Unstructured or Structured
 - Flat or Hierarchical
 - Uni-dimension or Multi-dimension
- Unstructured or Structured
 - Unstructured ... Gnutella, FreeNet, Winny, Skype
 - Structured
 - DHT base ... CAN, Chord, Pastry, Kademia
 - for Exact Matching
 - Non DHT base (Skip list base) ... Skip Graph, SkipNet
 - for Range Matching

24

まとめ

アプリケーション事例 (1)



買物支援サービスシステム「ワンダーキャッチャー」

- UAAプロジェクト(東大、NTT研、日立、阪大)にて開発
- 状況適合型コンテンツ推薦システム
- 青森県五所川原市のショッピングモール「エルムの街」にて一般の利用客を対象に実施
 - 期間: 1ヶ月(2006/2/4~3/5)
 - 約100店舗が参加、880コンテンツ
 - 参加人数: のべ2252人
 - ミューチップカード2000枚、マスコット端末数十台
 - 20分散端末へ広告コンテンツを発信
- PIAXを使って推薦システムを開発
性能、可用性、安定性について評価





2月6日付 東奥新聞1面

アプリケーション事例 (2)



MapWiki

- 共有地図を用いたユビキタスコンテンツ流通システム

- 共有地図上に誰でもコンテンツを追加可能
- ブラウザ上のリアルタイムな情報更新
- Wiki記法によるシンプルなリンク記述



Google Maps API を用いた非同期/リアルタイム編集・更新

編集モード

表示モード

発信 ↑ ↓ 共有

インターネット、携帯端末からの自由なコンテンツの発信と共有

- MapWiki on PIAX

- 口コミ情報共有サービスへの参加・離脱が容易に行なえ、自由なコンテンツ発信, アクセス制御が可能
- P2P アーキテクチャによりスケーラビリティ、規模拡張性を確保

これからやろうと思っていること

- 範囲探索機能の改良
 - ピアにおける複数位置のサポート
 - 返り値集約の機構改良
- 近接オブジェクトの発見
 - オブジェクト間の近接関係を処理する機構の導入
 - 統計処理: ベイジアン
 - 推論処理: 分散型演繹機構
- P2Pエージェントの移動透過性と複製透過性
 - P2Pエージェントの仮想化
 - 一群のP2Pエージェントを論理的に単一のP2Pエージェントとしてみせかける
- セキュリティ、プライバシー保護の強化
 - 現在、ピアとP2Pエージェントに対するPKIベースの認証機構のみ実装

29

まとめ

- 設計コンセプト
 - 近傍性、近接性と相互作用
 - ユビキタスP2Pサービスのコンセプト
 - P2Pエージェントとモビリティ
- 設計
 - アーキテクチャ
 - マルチオーバーレイ
 - P2PエージェントAPI
- 実装 – Skip Graph

について説明



PIAXの開発状況

- JDK5 以上のJVMで動作
- 2006年内にオープンソースとして公開
- GCJ(GNU Compiler for Java)を使ったARMデバイスへの組込み化もがんばっている

※PIAXの一部は、総務省委託研究「ユビキタスネットワーク認証・エージェント技術の研究開発」の一環として開発されたものです

30