

P2Pエージェントによる見守り・監視サービスの実現

寺西裕一, 田中博和, 石芳正
大阪大学サイバーメディアセンター

概要 P2P エージェントソフトウェアである PIAX を用いて見守り・監視アプリケーションを実装した。見守り・監視アプリケーションに要求されるセンサ情報配信の効率性、通知のリアルタイム性を確保するため、PIAX のマルチオーバレイ機能とディスカバリーメッセージング機能により、センサ側から監視者のピアを発見し、観測値更新イベントを通知する設計としている。これにより、センサ観測値を効率的に配信可能とするとともに、ブラウザ・携帯電話に双方向通信を行なう機能を組み込んだ軽量ピアを実装することで、リアルタイムな状態通知を可能とした。

1. はじめに

ユビキタス環境に散在して設置される監視カメラなどの映像を含む様々なセンサ情報をもとに、緊急事態等の発生を検知し、監視者やユーザに通知する見守り・監視サービスは、将来の安全・安心な社会を築く上で重要なアプリケーションに位置付けられる。こうしたアプリケーションでは、膨大に設置されたセンサから得られる情報から、監視者やユーザが必要とするものを効率的に選択・提示できる必要がある。

本稿では、こうした要求を満たすべく、P2P エージェントソフトウェア PIAX を用いて実装した見守り・監視アプリケーションについてその設計および実現法を述べる。

2. PIAX: P2P エージェントソフトウェア

我々の研究グループでは、ユビキタス環境において大量に存在するモノやセンサが発信する情報を効率よく扱うため、P2P のオーバレイネットワークとエージェントの機能を統合したプラットフォームとして 'PIAX' [1] を研究開発してきた。PIAX は複数のオーバレイネットワークを使い分けるマルチオーバレイ機能を持ち、分散ハッシュテーブル DHT、位置座標に基づく検索が可能な LL-Net、属性に応じたルーティングを行なう DOLR などが利用できる。また、ディスカバリーメッセージングと呼ばれる PIAX が持つリソース発見・連携機能は、P2P ネットワーク上の条件を満たすエージェントを動的に選択してメッセージを送信し分散処理を実行可能とする。本実装では、この PIAX を用いて見守り・監視アプリケーションを実現した。

3. 見守り・監視アプリケーション実現上の課題

ここでは、センサが分散して存在する状況を考慮し、P2P ネットワークにセンサがピアとして接続している環境を想定する。通常、見守り・監視アプリケーションでは、監視者やユーザはデスクトップのブラウザや携帯電話などのインタフェースを通じて情報を取得することが想定される。しかし、これらのインタフェースは、クライアント・サーバ型を想定しており、通常、クライアント側からサーバに接続し、能動的にデータを要求しなければ画面上に取得データを表示することができない。したがって、従来のアプリケーション構築手法において、継続的にセンサの観測値を得るには、一定時間おきにポーリングにより検索を行なうしかない。しかし、監視者側から一定時間おきに検索を行なってセンサ観測値を取得する方法には以下の問題がある。

- 1) 頻繁に更新されるセンサ情報配信の効率性
複数の監視者による取得要求が重なり、同時刻に検索が多数発生すると、同じ値を別の監視者からの要求に返却する処理がセンサピアに集中する。
- 2) 通知のリアルタイム性
実際のセンサ観測値の変化から、画面上に反映されるまでに、最悪ポーリング間隔(および通信遅延)にあたる時間の遅延が生じてしまう。

4. PIAX による 見守り・監視アプリケーションの実装

本節では上記問題点に対応すべく PIAX 上に実装した見守り・監視アプリケーションについて述べる。

4.1. センサ観測値のマルチオーバレイによる配信

本実装では、PIAX のマルチオーバレイ機能とディスカバリーメッセージング機能により、センサ側から監視者のピアを発見し、センサの更新情報を監視者ピアに配信する。ポーリングではなく、センサ側から観測値更新イベントとして通知を行なうことで、同時刻にクエリが多数発生した場合に同じ値を何度も返却する冗長性を排除できる。

監視者は監視対象となる領域の近辺にあるセンサの観測値を監視することを想定し、通常の状態では、センサから一定の距離を観測している観測者に対して通知を行なうこととした。具体的には、位置座標に基づくオーバレイである LL-Net を用いて、センサピアの周囲に存在するピアへ観測値更新メッセージを送出する。

一方、例えば火災などの異常が発生した場合、緊急状態として観測対象となる領域以外の位置に置かれたセンサの情報であっても、通知を得たいと考えられる。そこで本実装では、このような緊急状態の通知を効率的に行なうため、緊急状態を通知するための論理的なコミュニティを設けた。このコミュニティは DOLR オーバレイを用いて実装している(図 1)。

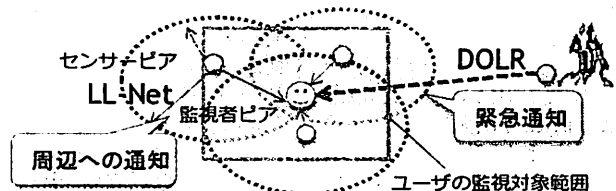


図1: オーバレイの使い分けによる配信

上記により通常状態は LL-Net、緊急状態は DOLR により配信というオーバレイの使い分けを行なっていることとなる。これは PIAX のマルチオーバレイ機能により実現されている。

4.2. 軽量 PIAX ピアの実装

ブラウザや携帯電話側からポーリングすることなく監視対象のセンサ観測値を得られるよう、任意のタイミングでリアルタイムにブラウザや携帯電話にメッセージを送信する軽量 PIAX ピア機能を実装した。

ここでいう軽量 PIAX ピアとは、以下を満たすピアを指す。

- ・ オーバレイのルーティングはそれ自身行なわず、通常ピア(親ピア)に依存
- ・ PIAX ネットワークとのインタフェースとしてディスカバリーメッセージングを利用可能
- ・ PIAX のエージェントとして動作可能

以下ではブラウザ、携帯電話それぞれについて軽量 PIAX ピアの実装方法を述べる。

4.2.1. ブラウザピア

DWR 等では Reverse Ajax と呼ばれる JavaScript のリモートプロシージャコールの仕組みが実装されている。これは、HTTP サーバ側の Java コードから任意のタイミングでブラウザ上の JavaScript を呼び出すための仕組みである。ブラウザからサーバへ張った HTTP セッションを張ったままとし、そのセッションを使ってサーバ側からプッシュを行なう方式を用いている。

この Reverse Ajax [2] を用いてブラウザ上に軽量 PIAX ピア機能を実現した。Reverse Ajax の HTTP セッションは、ブラウザから親ピアに張る。ブラウザのエージェントとしての動作は、JavaScript の関数(function)で記述する。図 2 は JavaScript による軽量ピアの動作記述例である。

```
// 軽量ピア動作開始
p = new LightPeer();
// 動作記述・属性設定
p.addMethod('hello', function(message) { alert('hello'); });
p.setAttribute('location', new Location(135.0, 35.0));
// 条件を満たすピアへのメッセージ送信
p.discoveryCall('location inside rect(134.5, 34.5, 1, 1)', 'hello');
// 軽量ピア動作終了
p.fin();
```

図 2: 軽量ピア記述例

4.2.2. 携帯電話ピア

ここでは、CLDC[2]/MIDP 準拠の J2ME が動作する携帯電話を対象とする。CLDC ではサーバソケットの規定があるものの、現在市場にある携帯電話では、サーバソケットが動作する J2ME 実装は少ない。また、通常クライアントソケットは利用することができるが、read と write を同時に行なうとアプリケーションが停止してしまう実装もある。このため、ブラウザと同様、サーバにあたるピアへセッションを張ったままとし、そのセッションを使ってサーバ側からプッシュを行なう方式を用いた。このときクライアントソケットは read 状態となるため、このセッションでクライアントにあたる携帯電話から任意のタイミングでデータの送付(write)ができない。そこで、データ送付時には専用のセッションを別に都度生成することで対処した(図 3)。エージェントとしての動作は基本的に J2ME により記述する。

本ピアの動作は Windows Mobile 6.0 が動作する携帯電話(PHS)上に組み込まれた JBlend (CLDC-1.1 MIDP-2.0) で確認している。

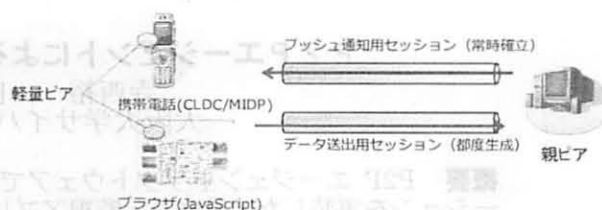


図 3: 軽量ピアにおけるメッセージ送受信

5. GoogleMaps を用いた地域監視アプリケーションの実装

以上述べた PIAX による見守り・監視アプリケーションとして、地図による地域監視アプリケーションを実装した。本アプリケーションでは、監視用の映像センサ(カメラ)が散在する環境において、地図上に表示される特定の領域を、ブラウザを用いて監視する。映像センサにおいて異常な状況(火災など)が発生すると、地図に表示されていない領域に存在する映像センサであっても、表示領域を当該センサの位置まで移動させて表示する。この動作はセンサから観測値が発信されたあと、ほぼリアルタイムで行なわれる。これらのブラウザ上の挙動は GoogleMaps を用いた JavaScript で記述している(図 4)。

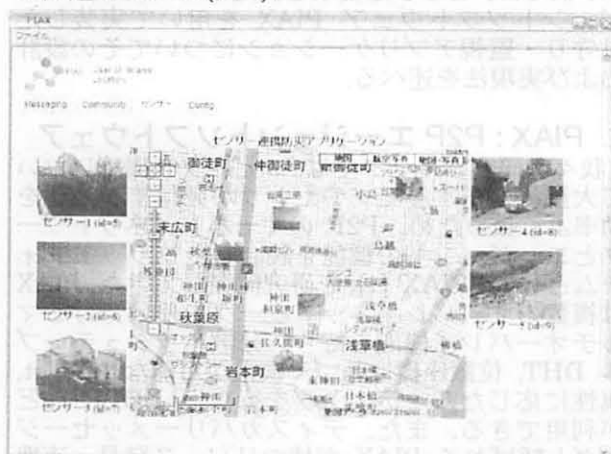


図 4: GoogleMaps による監視インタフェース(開発中の画面)

6. まとめ

本稿では、P2P エージェントソフトウェア PIAX を用いて実装した、見守り・監視アプリケーションについてその設計および実現法を述べた。今後は、よりトラヒックを削減することによる観測値通知メッセージ配信の効率化や、ピアの負荷を軽減する処理方式などについて検討をすすめる予定である。

参考文献:

- [1] 吉田幹, 寺西裕一, 春本要, 下條真司, 'マルチオーバーレイと分散エージェントの機構を統合化した P2P プラットフォーム PIAX', 情報処理学会研究報告 2006-DPS-128, pp.43-48, 2006.9
- [2] Joe Walker, Reverse Ajax, available at <http://getahead.org/dwr/reverse-ajax>
- [3] Java Community Process, JSR-000030 J2ME Connected, Limited Device Configuration, available at <http://jcp.org/aboutJava/communityprocess/final/jsr030/>