

ホスト情報を用いたユーザコンテキスト推測機構とその応用

坂本 憲昭^{†1} 西尾 信彦^{†2}

近年、ユビキタス環境において、コンテキストウェアコンピューティングの実現が重要視されている。これまで、様々なセンサや RFID タグなどを我々の生活空間に遍在させ、それらから取得したデータを解析することでユーザの状況を推測し、その状況に応じたサービスを提供する研究が多くなされてきた。一方我々の生活の中で、PC や PDA、スマートフォンといった小型・高性能化された端末を持ち運び、様々な用途で利用する機会が増加している。そこで本稿では、ユーザは必ず何らかの目的を持って端末を操作しているということに着目し、端末に関する情報のみを用いることによって、端末の利用目的に特化したユーザのコンテキストを推測するアプローチについて提案する。また、我々がこれまでに実装を行ってきたホスト情報を用いたユーザコンテキスト推測機構について紹介し、それを基に実装したサンプルアプリケーションについて述べる。さらに、ユーザの日常行動を基に端末の操作ログを取得し、コンテキスト推測を行った結果に対してポジティブマッチ・ネガティブマッチによる評価・考察を行い、本機構の有用性について検証した。

User Context Estimation Mechanism Utilizing Host Information and Its Application

NORIAKI SAKAMOTO ^{†1} and NOBUHIKO NISHIO ^{†2}

In recent years, on the occasion of coming ubiquitous society, to realize context-aware computing has been of importance. So far, to estimate user's context, many researchers have utilized various sensors or RFID in our living space. They have obtained various result from these devices and challenged to provide services that match the user's situation. On the other hand, mobile and smart devices have rapidly been carried in our daily life, such as PC, PDA and Smart Phone. It increases the chance to utilize these devices in various methods. Therefore, this paper paid attention to the tendency that such device's usage would reveal user's activity. We aimed at the user's context only using information related to these devices. We introduced our implemented system, user context estimation mechanism utilizing host information. The prototype system is designed and implemented with preliminary experiments. Furthermore, we evaluated our implemented system by the indicator, positive match and negative match, and varificated this system's utility.

1. はじめに

近年、ユビキタス環境において、ユーザの行動をシステムが解析し、その状況に応じたサービスを提供するための、コンテキストウェアコンピューティングの実現が重要視されている。これまで、様々なセンサや RFID タグなどを我々の生活空間に偏在させ、それらから取得したデータを解析することでユーザの状況(ユーザコンテキスト)を推測する研究が多くなされてきた。また、ユーザコンテキストに応じたサービスを提供するためのフレームワークやミドルウェアの研究も多く存在し、今後それらの研究が行われるにつれ、有益なコンテキストウェアサービスの提供が増加し

ていくことが予想される。しかし、外部的にセンサなどのデバイスを特別に持ち歩いたり、生活空間に配置させるには、コストやユーザの煩わしさなどの問題があり、未だに解決されていない。

一方で近年、PC や PDA、スマートフォンといった小型・高性能化された端末を持ち運ぶことが一般的となっている。ユーザは何気なく高性能な端末を持ち歩き、様々な用途で利用する機会が増加している。そこで本稿では、ユーザは何らかの目的を持って端末を操作していることに着目し、端末に関する情報のみを用いることで、ユーザのコンテキストを推測するアプローチについて提案する。また、我々がこれまで実装を行ってきた、ホスト情報を用いたユーザコンテキスト推測機構について述べる。

実現した機構では、外部アプリケーションを実装する際に API を通じてユーザコンテキストが取得できるため、様々なコンテキストウェアアプリケーション

^{†1} 立命館大学 理工学研究科 情報理工学専攻
Department of Computer Science, Ritsumeikan University

^{†2} 立命館大学 理工学部
Department of Science and Engineering, Ritsumeikan University

ンの実装が可能となる。本稿では、その実装例として、UPnP ライト制御サービスと連携させる実装を行い、ユーザコンテキストの変化によって自動的にライト制御を行うサービスの実現に成功した。また、システム自体の定量評価として、実際の日常生活を基にポジティブマッチ・ネガティブマッチを用いて評価を行い、その妥当性や有用性の検証を行った。

本稿は、全7章で構成され、2章では現在のコンテキストウェアの研究の現状とその問題点を列挙し、考察及び必要とされるシステムの要件を示した。また、3章では、これまでに提案を行ってきたホスト情報を用いたユーザコンテキスト推測機構についての実装概要について示し、4章では3章で紹介した機構を用いてサンプルアプリケーションを実装し、その定性評価と有用性の検証を行った。5章ではポジティブマッチ・ネガティブマッチを用いた評価を行った。6章では本研究の関連研究について述べ、7章ではまとめと今後について述べる。

2. コンテキストウェア研究の現状と問題意識

ユビキタスコンピューティングにおける「コンテキスト」という単語には、位置、時間、環境、状況等、ユーザの趣向や行動など様々なものが含まれ、我々コンテキストウェアに関する研究者は多様な手段を用いて、これらを推測しようとしている。このような各コンテキストをシステムが推測する手段として、カレンダーやリマインダツールなどのあらかじめユーザがシステムに通知しておく手法や、物理センサやインテリジェントセンサ、位置情報センサといったセンサデバイスの出力値を分析する手法が主流となっている。しかし、現状のコンテキストウェア研究では、いくつかの考慮すべき問題が存在する。

本章では、コンテキストウェアの研究に関するそれぞれの問題点について述べた上で、その考察を行い、本稿で提案するアプローチの要件について述べる。

2.1 コストやデバイス所持の問題

コンテキストウェアの研究で用いられる一手法として、RFID タグを日常空間のいたる場所や物に埋め込み、その対象物に対する人間の接触情報をキーとすることで、現在または未来の行動を的確に予想するという研究が行われている¹⁾。しかし、RFID 等の特別なセンサが実世界に埋め込まれ、かつ遍在する環境が訪れるには、まだ多くの時間がかかることが予想される。よって、我々の日常空間にタグが存在する想定環境を持つことで実現されるコンテキストウェアサー

ビスが実現するには、さらに多くの時間の推移が必要となる。

また、複数センサを有したデバイスを特別に身につけて生活を行うこと²⁾³⁾も、デバイスの大きさや、煩わしさの問題があり、ハードウェア的な解決がなされない限り、普及することは考えにくい。その裏付けとして、人にとって複数のデバイスを持って歩くための条件として、そのデバイスが有する特別な性能やメリットが必要であることや、出来る限り持ち歩く台数を減らしたいとの考えが働くという評価結果が報告されている⁹⁾。

2.2 異なるセンサの連携に関する問題

出力値が異なる様々なセンサが周囲に遍在している状況で、一つのコンテキストウェアサービスを提供することは、出力値の解析やサービス間の連携に膨大な計算コストがかかる。また、そのセンサに影響を与える個人によってもその出力値に大きな差が生じてしまうため、単一の閾値を設定するだけでは解決できない問題もあり、多くの課題が存在する。

これらの課題に関しては、コンテキストウェアの研究の中でも、センサフュージョン⁹⁾や、学習アルゴリズムなどの研究が盛んに行われているが、未だ発展途上なものが多く、実用적であるとは言い難い。

2.3 問題点からの考察及び要件

前述した問題点より、現在の環境下でコンテキストの推測を行う際には、新たにセンサなどを遍在させる手段をとるよりも、出来る限り既に普段持ち運ぶのが一般的となっているデバイスを用いての推測を行うことが望ましい。また、複数の外部センサからの情報を用いることなく、端末内部より得られる情報をユーザコンテキストを推測するためのセンサ情報として扱い、それらを解析することでユーザコンテキストを推測できれば、導入コストやユーザの端末を保持しておくことの煩わしさの問題も解消にもつながる。さらに、デバイス内部情報は、最もユーザの利用目的が反映されるため、その利用目的に応じたサービスの提供の実現が可能となる。

また、将来的に実装されるコンテキストウェアサービスには、コミュニケーションツールや推薦システムなど、様々なものがあると考えられる。よって、推測したユーザコンテキストを今後実装するアプリケーションから容易に扱えるためのインタフェースが必要となる。このようなインタフェースを用意することで、今後のコンテキストウェアや他分野のさらなる発展への貢献が期待できる。

3. コンテキストウェア推測アプローチの提案と実装

前章で示した問題点及び考察から、我々はユーザが普段利用している端末からの情報のみを用いることによって、ユーザコンテキストの推測を行う機構の提案と実装を行ってきた⁴⁾⁵⁾提案機構では、実行中のアプリケーションや、接続されているデバイス、その端末の位置などのOSが管理している情報（ホストコンテキスト）を用いることによって、ユーザのその端末の利用目的（ユーザコンテキスト）を推測することを目的としている。また、推測する主なユーザコンテキストの例として、「会議中にプレゼンテーションを聞いている」や「プレゼンテーションをしている」、「レポートを作成している」、「ただ Web ブラウジングをしているだけ」といったものが挙げられる。ホストコンテキストとユーザコンテキストの組み合わせの例を表1に提示する。表に示されている対応は、あくまで筆者特有の組み合わせであり、ユーザごとに異なることが容易に考えられるため、ホストを利用するユーザに応じて動的に変更できることが望まれる。

表1 トスとコンテキストとユーザコンテキストの組み合わせ例

ユーザコンテキスト	ホストコンテキストの組み合わせ
プレゼン発表中	プレゼン資料を操作している フルスクリーンモードで表示している プロジェクトが接続されている 会議室にいる
プレゼンを聞いている	チャットプログラムを使用している 議事録をとっている 音を出さないようにしている 会議室にいる
論文サーベイ中	Web ブラウザを使用している 検索作業が多い PDF ファイルを開いている
ゲーム中	ゲームのプロセスに対する操作 音楽プレーヤーが再生されている コントローラが接続されている
レポート作成中	エディタが開いている 描画ソフトを使用している Web ブラウザを使用している

このようなホストコンテキストの組み合わせのみにより、ユーザコンテキストの推測が出来れば、ホストの内部・外部問わず、様々なコンテキストウェアサービスの提供が可能となる。

以上のアプローチより、本章では、これまで我々が実装してきたホスト情報を用いたユーザコンテキスト推測機構の実装概要について述べる。本機構を構成する各モジュールの役割を示した後、実際にユーザコン

テキストを推測する手法について述べる。

3.1 モジュール構成

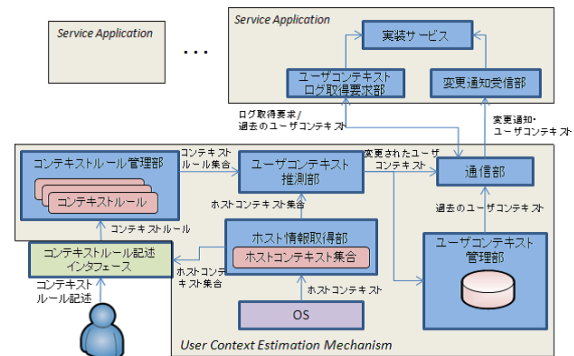


図1 システム全体図

本機構のモジュールは図1に示した通りである。

(1) ホスト情報取得部

ホスト情報取得部では、定期的にホスト上のOSから取得したリアルタイムなホストコンテキストを取得している。また、前回推測したホストコンテキスト集合を保持しており、取得したホストコンテキスト集合が前回のホストコンテキスト集合と異なる場合のみ、ユーザコンテキスト推測部及びコンテキストルール記述インタフェースへとその情報渡す。これにより、本機構は、ユーザのコンテキストが変更されたかどうか疑わしい場合のみ推測処理が行われ、全体的な処理にかかるコストの最適化を行っている。

(2) コンテキストルール管理部

本機構では、ホストコンテキストとユーザコンテキストの組み合わせをユーザが記述する。このことによって、各ユーザごとにカスタマイズした推測が可能となる。しかし、ユーザにとって一つ一つのルールを記述して行くには多大な労力が必要とされるため、その記述を簡略するためのインタフェースとして、コンテキストルール記述インタフェースを用意した。実装したコンテキストルール記述インタフェースを図2に示す。ユーザは表示されている現在、もしくは過去に観測されたホストコンテキストの中から複数の要素を選択し、ユーザコンテキスト名を記述し、保存する。

本稿では、特定のユーザコンテキストを記述し、XML形式で保存したものをコンテキストルールと呼ぶ。コンテキストルールは、構成する要素をXML形式で記述するものとした。その理由として、今後の汎用性を考慮している点が挙げられる。対象とするホストコンテキストの追加や、サービス制御などの機能追加に対

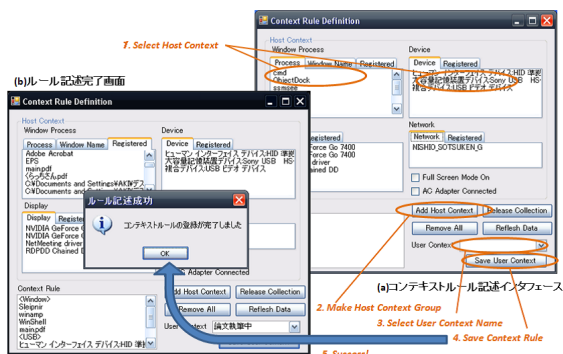


図 2 ユーザコンテキスト記述インタフェース画面

して拡張性を持たせるためである．コンテキストルールの記述例を図 3 に示す．

```

<ContextRule>
  <Context>
    <ContextName>論文執筆中</ContextName>
    <Window>
      <item>1. music collection</item>
      <item>Sleipnir</item>
      <item>WinShell</item>
    </Window>
    <USB>
      <item>ヒューマン インターフェイス デバイ
        ス:HID 準拠マウス</item>
    </USB>
    <Display />
    <ESSID>
      <item>0016017F01C2</item>
    </ESSID>
    <FullScreen>False</FullScreen>
    <ACAdapter>True</ACAdapter>
  </Context>
  <Service>
    <Enable>Light Control On</Enable>
    <Disable>Volume Control Off</Disable>
  </Service>
</ContextRule>

```

図 3 コンテキストルール例

コンテキストルール管理部では、これらのコンテキストルールをコンテキストルール集合としてまとめて保持し、ユーザコンテキスト推測部からの要求に対してルール情報を提供する．

(3) ユーザコンテキスト推測部

ホスト情報取得部より渡された、リアルタイムなホストコンテキスト集合を取得する．また、コンテキストルール管理部より取得した、コンテキストルール集

合とホストコンテキスト集合を用いてユーザコンテキストを推測する（推測の手法は後述）．

ユーザコンテキスト推測部は、前回推測したユーザコンテキストを保持しており、推測されたユーザコンテキストが前回のユーザコンテキストと異なる場合のみユーザコンテキスト管理部及び通信部に更新情報として送る．

(4) ユーザコンテキスト管理部

ユーザコンテキスト推測部から送られてきたユーザコンテキストは、ユーザコンテキスト管理部でログとして保存される．このとき、ユーザコンテキスト名と共に、推測した時間も格納される．ユーザコンテキスト推測部では、通信部から要求された対象時間のユーザコンテキストを選択し、そのコンテキスト名や、コンテキストの期間などを渡す．

(5) 通信部

通信部では、ユーザコンテキスト推測部において変更が認識された際に、本機構を用いて実装されたアプリケーションに対して変更通知を行う．また、アプリケーション側から過去のユーザコンテキストの取得要求も受け、ユーザコンテキスト管理部より対象時間のユーザコンテキストを取得し、アプリケーションに受け渡す役割りも果たしている．アプリケーション側からのこれらの要求は、通信部で用意した API を用いて容易に処理が可能となっている．

3.2 ユーザコンテキストの推測

ホスト情報取得部によって、対象ホストコンテキストの変更が確認された場合に取得されるホストコンテキスト集合は、コンテキストルール管理部が保持するコンテキストルール集合と比較される．そして、コンテキストルール集合の中から最も類似するコンテキストルールが自動的に適応され、ユーザコンテキストの推測結果となる．本稿では、この推測処理のことをコンテキストマッチングと呼ぶ．コンテキストマッチングの処理の流れを図 4 に示す．

最も類似するコンテキストルールは、以下の条件に基づいて選択される．

- ホストコンテキストとコンテキストルール内の要素が最も多い
- 現在最もアクティブなホストコンテキストを含む
しかし、上記の条件を適用しても複数のコンテキストルールが選択される場合がある．その場合は、ユーザに対して再度ルール記述要求を行い、明確な違いを含んだ記述を要求する．

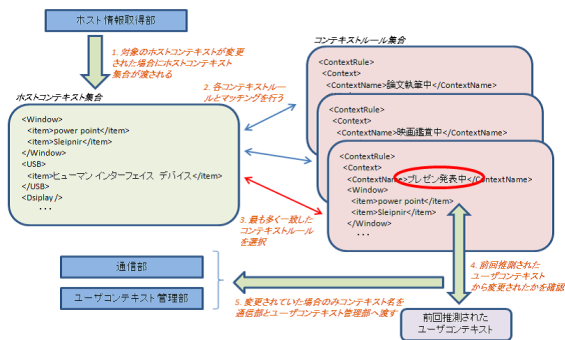


図 4 コンテキストマッチングの流れ

4. サンプルアプリケーションの実装

4.1 実装環境

本稿では、前章で示した機構を用いたサンプルアプリケーションの実装を行った。対象としたサービスは、スマート環境下に存在するライト制御サービスである。ユーザがコンテキストルール記述を行い、そのルールのコンテキストが推測された場合に実行するサービスを同時に記入するためのインタフェースの追加実装を行った。また、本システムの通信部のメソッドを用いることで、ライトサービスサーバへの通信を行うアプリケーションも作成した。作成したアプリケーションでは、ルールに合致したユーザコンテキストが推測された場合に、ライトの ON/OFF の制御要求をサーバに伝える。

サービスアプリケーションの構成環境を図 5 に示す。ライトサービスアプリケーションは、UPnP 準拠で作成されたものであり、制御を行うサーバ及びライトサービスは PLC で通信されている。本サービスアプリケーションの実装を行い、「プレゼンテーションを開始した場合にライトを On にし、終了したときにライトを Off にする」というルール記述を行った。実際に使用し、ルール記述内容に準拠した処理が行われることを確認した。

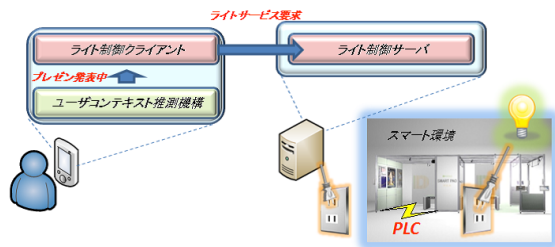


図 5 ライト制御構成

4.2 サンプルアプリケーションの評価と考察

実装したアプリケーションは、個人でスマート環境にあるデバイスに対して要求を出すのみならば、問題なく動作することが確認された。しかし、複数人で使用した際には、排他制御の機能がなく、後から操作要求を出したユーザに権限が移ってしまうといった問題が見受けられた。

一般にスマート環境でのユーザコンテキストに基づくサービス構築の際、考慮しなければならない事項として、サービス利用者の要求の競合が挙げられる。例えば、ユーザ A からの「映画を見る時にライトを消す」という要求と、ユーザ B からの「部屋に入ったときにライトをつける」といった要求が同時に発生した場合に、どちらの要求をサービスサーバが受け入れるかといった、要求の取捨選択の基準を定めておく必要がある。

この問題に対して、マルチレベル優先順位ポリシー (MPDP) を定義することで、スマート環境下に存在するサービスごとにユーザによる優先順位を決定するという研究がなされている⁷⁾。本サービスアプリケーションの実装では MPDP の導入を行っていないが、問題の解決策として提案のみ行っておく。

5. 評価

5.1 評価方針

本稿で行う評価方針としては、本システムを動作させた状態で普段と同様の生活を行うことで評価用データを収集する。その後、実際のユーザコンテキストと推測したユーザコンテキストを比較し、純粋なコンテキスト推測のみの精度を求める。比較の指標はポジティブマッチ・ネガティブマッチを行うこととする。以下にその評価手段の詳細について述べる。

5.1.1 評価用データの取得

評価要件

前述した評価方針を取ろうとすると、実際に正確なユーザコンテキストを取らなければならない。しかし、現状正確なユーザコンテキストを取得できるツールは存在しないため、ユーザ自らがホストを使用する際のコンテキストを平行で逐一時間と共に記録していくことで、それを実際の正確なユーザコンテキストとし、システムが推測したユーザコンテキストとの比較対象とする。

本システムは通常、ホストコンテキストに変更があった際にユーザコンテキストの推測も行っている。また、二つ以上のコンテキストを推測した場合にはユーザに対して再ラベリングするように要求するインタフェー

スの動作も行うことで、ルールが精練されていく機能も存在する。しかし、本稿で行う評価として、純粋な推測のみの評価値を求めることを目的としているため、これらを分離する必要がある。

機能分離詳細

前述した要件により、本システム内の各モジュールの分離を行う。まず初めに、純粋なホストコンテキストログのみを取得するために、ホスト情報取得部のみを単独で動かすよう実装変更した。ホスト情報取得部は、ホストコンテキストに変更があった場合には、ユーザコンテキストの推測処理は行わず、別に用意したホスト情報ログ DB への保存のみを行うようにした。また、ユーザインタフェースへの表示も行わず、常にバックグラウンドでのみ動作するものとした。

以上のシステム変更を行った上で一日の生活を行うことでホストコンテキストのみのログを取得した。その後、ユーザコンテキスト推測部において、取得したホストコンテキストログデータを用いて一括で推測処理を行った。本システムで実装したユーザコンテキスト推測部は本来、リアルタイムなホストコンテキストをホストコンテキスト取得部から取得しての推測を行うが、評価のために、ホスト情報ログ DB への参照を行い、DB に格納されている全てのホストコンテキストに対して推測処理を行うように実装変更した。また、推測処理を行うためのルールに関しては、ログデータの取得を行う前に、あらかじめユーザの行動を予測した上で記述しておくものとする。

5.2 評価指標の定義

以下に本稿で行う評価の指標について述べる。本稿で用いる指標として、ポジティブマッチ・ネガティブマッチでの評価を行う。本手法は Inductive Logic Programming System⁽¹⁰⁾ などのデータマイニングの研究でも一般的に用いられており、コンテキストの精度を求める指標として有用であると言える。

ポジティブマッチ

本稿で述べるポジティブマッチとは、どのくらいの割合で実際のユーザコンテキストをシステムが推測出来ているかを表すための指標である。ポジティブマッチによる評価結果が高い場合は、そのシステムが行う推測の精度が良いことを示す。式中の T_i は、所々に現れる実際のユーザコンテキスト A が推測したユーザコンテキストと一致した場合の時間数を示す。 T_{All} は、評価時間中のすべての T_i の総和であり、ユーザコンテキスト A のポジティブマッチによる評価値である PM_A は、 T_{All} を実際のユーザコンテキスト A の総時間で割った値となる。以下の式にすべてのユーザ

コンテキストを当てはめ、ポジティブマッチとして評価を行う。

$$PM_A = \frac{T_{All}}{SR_A} \times 100(\%) \quad (1)$$

$$T_{All} = \sum_{k=1}^n T_i \quad (2)$$

T_i → 実際のコンテキスト A が推測したコンテキストと一致した時間数

SR_A → 実際のコンテキスト A の総時間数

ネガティブマッチ

本稿で述べるネガティブマッチとは、システムによって推測されたユーザコンテキストが、実際のユーザコンテキストと異なる確立を求める指標である。ネガティブマッチに関しても、評価結果が高い場合は、そのシステムの推測の精度が良いことを示す。つまり、推測したユーザコンテキストは正確な場合にのみ推測されていることである。式中の T_i は、所々にシステムによってユーザコンテキスト A と推測された時に実際のユーザコンテキストと比較して一致していた時間数を表わす。また、 $T_{N_{All}}$ はその総和であり、ユーザコンテキスト A のネガティブマッチによる評価結果である NM_A は、 $T_{N_{All}}$ を推測したユーザコンテキストの総時間数 SE_A で割った値を百分率から除いた値である。

$$NM_A = (1 - \frac{T_{N_{All}}}{SE_A}) \times 100(\%) \quad (3)$$

$$T_{N_{All}} = \sum_{k=1}^n TN_i \quad (4)$$

TN_i → 推測したコンテキストが実際のコンテキストと一致しない時間数

SE_A → コンテキスト A を推測した総時間数

図 6 にポジティブマッチ、ネガティブマッチの計算例を示す。

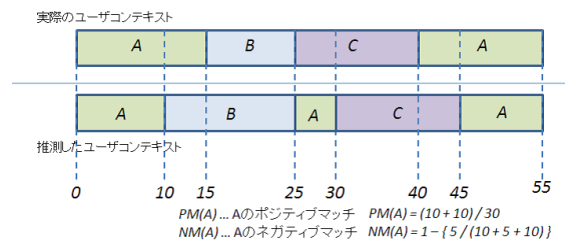


図 6 ポジティブ・ネガティブマッチ例

5.3 全体評価

全体的なユーザコンテキスト推測の評価として、すべてのユーザコンテキストに対するポジティブマッチ

を行う。つまり、システム稼働中に推測したユーザコンテキストがどの程度の精度で正しい推測が行われているのかを評価する。また、全体的なポジティブマッチであるため、ネガティブマッチはポジティブマッチと等しくなるため、同一のもののみならず、評価に用いるためのデータを図7に示す。以下、このデータを基に評価を行うものとする。

【推測結果】		【実際の行動】			
CONTEXT_NAME	DATETIME	CONTEXT_NAME	DATETIME	場所	
:	:	:	:		
プログラミング中	2008/2/14 13:03:40	プログラミング中	2008/2/14 13:03:36	研究室	
プレゼン作成中	2008/2/14 13:15:34	プレゼン作成中	2008/2/14 13:15:32		
メールチェック中	2008/2/14 13:28:40	メールチェック中	2008/2/14 13:28:20		
プレゼン作成中	2008/2/14 13:45:19	プレゼン作成中	2008/2/14 13:45:19		
プレゼン発表中	2008/2/14 14:01:17	プレゼン発表中	2008/2/14 14:11:20		
プレゼン作成中	2008/2/14 14:28:43	論文執筆中	2008/2/14 14:28:10		
論文執筆中	2008/2/14 14:30:10	プログラミング中	2008/2/14 15:03:22		
プレゼン作成中	2008/2/14 14:56:40	先生とチャット中	2008/2/14 15:33:58		
論文執筆中	2008/2/14 15:02:43	プログラミング中	2008/2/14 15:40:00		
プレゼン作成中	2008/2/14 15:02:49	レポート作成中	2008/2/14 15:54:28		
プログラミング中	2008/2/14 15:02:58	クイズとチャット中	2008/2/14 16:16:10		
先生とチャット中	2008/2/14 15:34:43	SNSのデータベース確認	2008/2/14 16:26:40		
プログラミング中	2008/2/14 15:37:04				
チャット中	2008/2/14 15:42:31	プログラミング中	2008/2/14 16:58:30		フォレストハウス
先生とチャット中	2008/2/14 15:42:34	シートダウン	2008/2/14 18:16:15		
プログラミング中	2008/2/14 15:54:28				
レポート作成中	2008/2/14 15:54:59				
プレゼン作成中	2008/2/14 15:56:52				
:	:	:	:		

図7 評価用データの一部

$$PM_{All} = 84.4(\%)$$

全体評価考察

全体の精度としては以上のような値が算出された。評価を行う際、ユーザが逐一自分の現在の状態を記述していくが、その記述時間のずれによるノイズが発生するため、評価値に若干の影響を与えたことが実データから推測出来る。また、その他のノイズとして、ユーザコンテキストが変更される際に、センサ値であるプロセスや、デバイスのON/OFFの移り変わりが頻繁な場合に、出現するはずのないユーザコンテキストが推測されてしまう場合がある。コンテキスト間での移り変わりのときに、そのような出現するはずのないコンテキストをノイズとして除去する機構があれば、より高精度な評価値が得られる可能性がある。

その他にも、別の作業の合間に目的の操作を行っている場合などに問題があると考えられる。例えば、ユーザにとってある時間のコンテキストが「Aさんとチャット中」といったユーザコンテキストの場合を想定してみる。実際にチャット画面に対して操作を行っている場合はその通りにシステムによって推測されるが、チャットの返信を待つ間にレポートを作成する画面に対して操作を行った時に別コンテキストとして推測されるといった場合もあった。このような事例は、日常的によく見られるため、行動パターンを学習するようなアルゴリズムの導入などの対策を考える必要がある。

しかし、本稿で用いたような簡単なアルゴリズムを用いただけでも、ユーザコンテキストの種類が多く、

頻繁に移り変わっていくような環境の中で、十分な精度で推測を行うことができたと言える。

5.4 個別評価

前節ではシステム全体の精度の評価であったが、本節では各コンテキストに応じてポジティブ・ネガティブマッチによる評価を行った。その評価結果の中より、特徴的なものに焦点を絞って示し、その考察について述べる。表2に代表的なユーザコンテキストにおける評価結果を示す。

表2 個別評価結果

CONTEXT_NAME	Positive Match	Negative Match
プレゼン発表中	96.6%	56.9%
プログラミング中	99.9%	93.5%
論文執筆中	83.4%	98.2%

個別評価考察

表2に示したように、記述したルールによってかなりの差が生じた。プログラミング中では、ポジティブマッチ・ネガティブマッチ共に90%以上の精度を記録した。しかし一方、プレゼン発表中では、ポジティブマッチこそ95%以上の精度を記録したが、ネガティブマッチは57%程度であった。また、論文執筆中はプレゼン発表中とは逆に、ネガティブマッチは98%の精度であったが、ポジティブマッチでは83%程度にとどまっている。

以上の結果より、ルールの記述方法によって大きく精度結果に差が生じるという推測が行える。プレゼン発表中でのネガティブマッチの結果が低いという結果は、推測してはいけない場合の推測が多いということである。つまり、プレゼン発表中のルールに記述されているホストコンテキストは、その他のルールでも同様に記述されている場合が多く、プレゼン発表中ルールの方がそれらルールよりも影響力が強いことが推測される。一方、論文執筆中のように、ポジティブマッチの精度が低い場合、ルールの記述に不備があるか、先ほどのプレゼン発表中のような影響力の強いコンテキストルールに引っ張られている可能性がある。

これらの考察から、影響力の強いコンテキストルールの存在が確認でき、そのルール中には他のルールでよく記述対象となるようなホストコンテキストが多く記述されているという推測が行える。逆に言えば、そのようなよく記述されるようなホストコンテキストの重みを減らすなどの重みづけの処理を行って比較することで、それぞれの制度の改善が期待できる。

6. 関連研究

利用者の状況と好みに基づいた適切なコミュニケーションサービスを選択する手法の提案

慶応大学村井研究室の奥村氏の研究であり、近年のコミュニケーションサービスの増加に伴い、コミュニケーションの受信側でその手段の取捨選択を行うことが出来ない事を問題点として挙げている⁸⁾ 受信者の状況に“最適”なコミュニケーションサービスを決定するために、キーボードやマウスの入力イベントとその入力先アプリケーション情報から状況を類推し、利用者の好みに基づいて最適なコミュニケーションサービスを選択するといったアプローチを取っている。また、ユーザ状態における受信したいサービスを指定するための各ユーザが設定するプリファレンスの提案を行っている。評価結果として、端末を継続的に利用するシナリオでは97.4%といった高い精度でユーザのニーズに見合ったコミュニケーション手段が選択されている。

本稿で提案するホストコンテキストを用いてユーザコンテキストを推測するといったアプローチが類似した研究である。しかし、本稿で実現した機構では、コミュニケーション手段の選択のみに着目しておらず、コンテキストウェア全般のサービス開発のためのミドルウェアとしての機能を果たす目的であるため、異なる。

7. まとめと今後の研究活動

本稿では、始めに現状の多くのユーザコンテキスト推測に関する研究で見られるような、RFIDなどのタグや外部センサを用いる手法の問題点について述べた。また、その問題点より研究方針についての考察を行い、単一ホストの内部情報をユーザコンテキスト推測のためのセンサ情報とすることでのアプローチの有用性とタグ情報などから推測出来ないような、ユーザの端末操作の目的の推測が可能となることを期待できることについて述べた。また、我々がこれまで提案を行ってきたホスト情報を用いたユーザコンテキスト推測機構について説明し、実際にそのコンテキスト推測機構を用いてライト制御サービスの実装を行い、その有用性を確認した。

評価では、ポジティブマッチ・ネガティブマッチを用いた評価を行い、全体では84.4%の評価値を得られ、コンテキスト個別では56.9~99.9%といった幅広い結果が得られたと同時に、ルール記述ごとに影響力があり、強いコンテキストルールに引っ張られてしまう可能性が判明した。

今後の研究活動では、ルール記述の影響力を均一化するために、ユーザがルールを記述した際に、自動的に重み付けを行っていくことで、評価値にどのような影響がでるのかを検証していく。また、有効な推測アルゴリズムを適応させることで、コンテキスト推測精度の向上を図っていく予定である。また、近年の端末に付属してきた加速度センサやWebの閲覧情報などの多様なホストコンテキストを用いて、推測可能なユーザコンテキストの範囲も広くしていきたいと考えている。

参考文献

- 1) 楓仁志, 山原裕之, 野口豊司, 島田幸廣, 島川博光, “接触物体から個人の行動を認識するための効率的な手法”, 情報処理学会論文誌, Vol.48, No.3(20070315) pp.1479-1490.
- 2) Steve Mann, “Wearable Computing: A First Step Toward Personal Imaging”, Computer, Vol.30, No.2, pp.25-32.
- 3) Jonathan Lester, Tanzeem Choudhury, Gaetano Borriello, “A Practical Approach to Recognizing Physical Activities”, In Proceedings of The Fourth International Conference on Pervasive Computing, pp.1-16, 2006.
- 4) 坂本憲昭, 倉田哲也, 西尾信彦, “ホスト情報を用いたユーザコンテキスト推測機構”, DI-COMO2007.
- 5) 石原孝通, 倉田哲也, 久野綾子, 西尾信彦, “PCはどのくらい自位置に近づけるか?”, 新世代ロケーションウェア技術・サービスに関するワークショップ (Place+).
- 6) Global Information, Inc. “世界のモバイルデバイス市場”.
- 7) 米澤拓郎, 守分滋, 高汐一紀, 徳田英幸, “マルチユーザ・コンテキストウェア・サービス環境のための柔軟な機器排他制御フレームワーク”, SPA2005.
- 8) 奥村祐介, 小原泰弘, 南正樹, 村井純, “利用者の状況と好みに基づいた適切なコミュニケーションサービスを選択する手法の提案”, DICOMO2007.
- 9) Eric Horvitz, Paul Koch, Carl M. Kadie, Andy Jacobs, “Coordinate: Probabilistic Forecasting of Presence and Availability”, Microsoft Research.
- 10) Raymond J. Mooney, Prem Melville, Lappoon Rupert Tang, Jude Shavlik, Ines de Castro Dutra, David Page, Vitor Santos Costa, “Relational Data Mining with Inductive Logic Programming for Link Discovery”, Proceedings of the National Science Foundation Workshop on Next Generation Data Mining, Nov. 2002.