

効果的なライフログ閲覧のための柔軟な配置と 協調表示可能な可視化システム

浜地 亮輔^{†1} 北沢 匠^{†1} 榎堀 優^{†2}
新井 イスマイル^{†3} 西尾 信彦^{†4}

携帯端末から得られる位置情報やセンサ情報などのライフログを蓄積・解析することで、ユーザの生活習慣分析や行動予測を行うといった研究が盛んに行われている。このような研究を行う際には、センサデータの解析やマイニングプログラムのデバッグといった工程が発生する。これらの工程において研究者はデータを可視化する必要がある。場合によっては可視化システムそのものを開発しなければならず、負担となる。また、既存の可視化システムで複数のデータを可視化した場合、それぞれのグラフが独立しているため、研究者自身がデータの対応付けを目視で行う必要があり、効率が悪く、本研究では、そのような負担を軽減するためのライフログ可視化システムを実装した。このシステムでは、データベースに格納されたデータならば基本的に全て、コードを書くことなく可視化できる。また、時刻を指定できるスライドバーによって複数の可視化結果を協調させ、スライドバーの位置に対する時刻のデータをグラフ上に示すことで、データの対応付けを容易にした。本システムで評価実験を行った結果、より効果的に複数のデータを対応付けできることを確認した。

Visualization System with Flexible Arrange and Collaborative Movement for Lifelog Analysing

RYOUSUKE HAMAJI,^{†1} TAKUMI KITAZAWA,^{†1}
YU ENOKIBORI,^{†2} ISMAIL ARAI^{†3}
and NOBUHIKO NISHIO ^{†4}

Recently, we have been able to get and accumulate a variety of data such as location information and acceleration from mobile phones. There is an increasing amount of research on Mining to analyze lifestyle and predict action using accumulate data. These researches have progress of the work such as analyzing lifelog and debugging Mining program. In this work, Mining researchers have to visualize lifelog and result data of Mining. Sometimes the

researchers have to make a program for visualization. It becomes heavy burden for them. In addition, it is not efficient using current visualization applications if researchers compare some data at the same time because they have to make sure correspondences among data. Hence, we implemented a graph generation application that helps to analyze lifelog data and debug Mining programs to reduce workload of visualizing data for the researchers. This system can visualize all lifelog and Mining results put in database. The researchers become able to visualize date without writing some source code. We put the slide bar which is adjusted time on page to achieve the mechanism that moves tooltips dynamically in two or more graphs. The result of the experiment prove that the slide bar in this system is useful.

1. 序 論

近年の携帯端末の高機能化は目覚ましく、GPSなどの測位デバイス、Wi-FiやBluetoothなどの無線通信デバイス、加速度センサやジャイロセンサなどのセンサデバイスが搭載されるようになってきた。これらのデバイスから取得できるデータの中には、携帯端末を持ち歩く人間の日常生活に関連しているライフログが含まれている。ライフログを蓄積・マイニングすることで、持ち主の生活習慣を推測したり、更にその推測結果から行動予測を行うなどといった研究が盛んに行われている。

例えば、KDDI研究所¹⁾は、日常生活において、ライフログを蓄積し、管理するシステムである、ケータイ de ライフログ^{*1}を研究開発している。ケータイ de ライフログは、蓄積されたライフログをマイニングすることで、ライフログ提供者の趣向や予定を推測し、以後の行動で有益と考えられる情報を推薦するサービスや、リマインドを送るサービスなどを提供する。森岡ら²⁾は、携帯端末に搭載されている加速度センサや気圧計のデータを事前に蓄積・マイニングし、移動中の携帯端末上で得られたデータとその場で照合することで、「歩く」「走る」といったユーザの状態を推定している。また、太田ら³⁾は、Wi-Fi基地局の観測履歴を解析した結果と、現在のWi-Fi基地局観測データを照合することで、ユーザが今どこに停留しているか、もしくはどこを移動中かを推定している。

^{†1} 立命館大学 情報理工学部 情報システム学科

^{†2} 名古屋大学 大学院情報科学研究科

^{†3} 明石工業高等専門学校 電気情報工学科

^{†4} 立命館大学 情報理工学部

*1 <http://www.kddi.com/business/oyakudachi/square/labo/003/index.html>

以上のようなマイニングの実現には、携帯端末から取得したライフログの分析や、マイニング結果の妥当性判断などが必要となる。しかし、ライフログやマイニング結果を数値の羅列として参照し、その変化やデータ同士の対応を判断することは難しい。よって、それらのデータをグラフ化したり、地図上にアイコンとして表示したりする可視化ツールを作成する手法が一般的である。しかし、このような工程はマイニングそのものに注力したい研究者にとって負担であり、開発速度低下の要因となっている。

そこで本論文では、データマイニングを行うプログラム(以下、マイニングプログラム)の開発における負担を軽減するために、ライフログ及びマイニング結果を少ない手順で可視化し、効率的に参照できるシステムである DUBIX を提案する。DUBIX は、データベースから取得できるデータならば基本的に全て可視化できる。また、データを可視化する際に、対象データのスキーマに応じた適切なグラフをシステムが判断する。ユーザは描画したいデータとその条件を GUI の操作で指定するだけで、ライフログやマイニング結果をフィルタリングして可視化できる。さらに、複数の可視化結果を同時に閲覧可能にし、時間を基準としたスライドバーと描画した各グラフを対応付けて、スライドバーの位置と対応した時刻におけるデータを各グラフ上に示することで、より効率的な解析を行えるようにした。本論文は全 6 章で構成されている。第 2 章では、ライフログを利用するマイニングプログラム開発における可視化作業の問題点を述べる。第 3 章では、問題点に対する要件とアプローチを明らかにする。第 4 章では、本システムの実装について紹介する。第 5 章では、本システムの評価と、それに対する考察を述べる。第 6 章では、まとめと今後について述べる。

2. マイニングプログラム開発における可視化作業の問題

GPS 情報や加速度センサの値などの携帯端末から得られるライフログは、数値あるいは記号で保存されるため、マイニングプログラム開発者はそれらの可視化を行うことが多い。また、多くの場合、複数のセンサデータを利用して、それぞれの可視化結果を同時に見比べることで、データの対応を参照する。例えばユーザが一定時間留まっていた位置を調べる場合、開発者は地図上に表示されたユーザの位置情報とユーザの速度のグラフを見比べ、照合することになる。さらに、開発者は作成したマイニングプログラムが適当であるか確認するためのデバッグが随時必要になる。マイニング結果もセンサデータと同様に数値か記号で表されることが一般的であり、センサデータと同様にグラフや地図を用いた可視化を行うことが多い。このような要因から、マイニングプログラム開発者は可視化システムを自作するか、既存のものを利用する必要がある。

2.1 マイニングプログラム開発者による可視化システム作成における問題

ライフログや、マイニング結果を可視化するためのシステムを一から開発するには時間がかかる。また、Google Chart Tools^{*1}などのデータ列を簡単に可視化するためのライブラリがあるが、それを用いた追加のプログラムの開発が必要である。したがって、マイニングプログラム開発者は可視化システムの作成やデバッグに時間が割かれ、本来の目的であるマイニングプログラムの開発が遅れる要因となっている。

2.2 既存の可視化システムにおける課題

現在、数値データを可視化するための、様々なシステムが提供されている。堀之内ら⁴⁾は、地球流体のデータを可視化・解析するために、GfdNavi を開発した。地球流体データは非常に複雑であり、可視化するためには事前に保存されているデータを処理する必要がある。そこで、研究者のほぼ全員が利用している GPhys^{*2}と呼ばれるライブラリを用いて実装を行った。また、アプリケーション内で GPhys を利用したコードを記述し、アプリケーションの処理結果を描画する機構も実現した。

Microsoft Office Excel^{*3}や Google ドキュメント^{*4}、OpenOffice.Org^{*5}などのスプレッドシートアプリケーションには、グラフを作成するための機構が存在する。利用者はあらかじめ表に可視化したいデータを記入するか、データベースやファイルから読み込んでおく。そのデータの可視化したい範囲を選択し、グラフの種類や x 軸、y 軸の指定などを行うことによって簡単にグラフを描画できる。また、OpenOffice.Org は拡張機能である Euro Office Map Chart^{*6}を用いることで、地図上へのデータの描画も行うことができる。マイニングプログラム開発において、これらの可視化システムを用いた場合の問題点を述べる。

容易に多様なフィルタリングができない

堀之内らのアプリケーションは、様々なフィルタリングを行うことができない。また、スプレッドシートアプリケーションはフィルタリングを行うことはできるが、可視化対象となるデータを表に打ち込む作業に手間がかかる。また、データの種類に応じてどのグラフを描画すればよいかの判断は利用者が行わなければならない。

*1 <http://code.google.com/intl/ja/apis/chart/>

*2 <http://ruby.gfd-dennou.org/products/gphys/>

*3 <http://office.microsoft.com/ja-jp/excel/>

*4 <http://www.google.com/google-d-s/hpp/hpp-ja-jp.html>

*5 <http://www.openoffice.org>

*6 http://www.multiracio.com/index.php?style=eurooffice&page=eo_ext_mapchart

各グラフが独立状態である

スプレッドシートアプリケーションは、ひとつの画面内に複数のグラフを生成することができる。複数のグラフを描画し、比較しやすい位置に配置することで、利用者は各グラフに描画されたデータを対応付けることが容易となる。しかし、多くの場合それらのグラフは独立しており、データの対応付けは目視によって行われる。これでは同時に複数のグラフを描画しても、利用者は各グラフの関係を理解するのに時間がかかる。例えば、移動経路と速度の変化を知りたい場合、時間を軸としたグラフ(以下、時相グラフ)と、データを地図上に表示するグラフ(以下、位置グラフ)を用いる。しかし、これらをただ単に描画しても、ある時間内の移動経路と速度の変化が示されるだけで、ある地点では速度の変化が激しいといった情報を得るには、それぞれのグラフで時間を目視で照合する必要がある。

3. 要件とアプローチ

3.1 本研究の要件

コードを書かずに少ない手順でグラフを描画

可視化を行うための新たなプログラムを開発することで、そのプログラムのデバッグも必要となり、開発者の負担が増えてしまう。したがって、マイニングプログラム開発者の負担になるようなプログラミングを極力必要とせず、少ない手順で、グラフを描画できることが望ましい。

多様なフィルタリングへの対応

ライフログやマイニング結果の解析には、それらをただ可視化するのではなく、様々なフィルタリングを行う場合が多い。フィルタリングを行うことで、マイニングプログラム開発者はより抽象的なユーザの行動を参照できる可能性がある。したがって、多様な種類のフィルタリングに対応するべきである。

利用できるグラフ形式の自動判定

可視化したいデータの形態によって、描画できるグラフは異なる。ライフログやマイニング結果の多くは時間を基準にして保存されているので、時相グラフを描画できる。また、ライフログやマイニング結果の中には、GPSによって得られる位置座標データを含むものがある。この場合、データを位置グラフとして描画することができる。このように、描画したいデータがどのようなグラフで描画できるかを、ユーザが判断するのではなく、システム側で判定するような機構が望まれる。

複数のグラフの協調動作

1種類のデータのグラフだけでは得られる情報は少ないため、ユーザの行動をより詳細に分析するためには、複数種類のデータのグラフを照らし合わせる場合が多い。しかし、描画された複数のグラフを並べて、目視で照合するのは煩わしく、ミスも発生しやすい。したがって、複数のセンサデータを同時に描画可能であることだけでなく、注目点の移動を複数のグラフに対して同時に行う、グラフ同士を協調動作させることが可能な機構であることが求められる。

グラフの描画位置指定

複数のグラフを同時に描画した場合、分析の目的によってグラフの適切な描画位置は異なる。例えば、複数のグラフが描画された状態で、位置情報と加速度を比較する場合、その2つのグラフが離れた位置に配置されていると、利用者はそれらの関係を把握しにくくなる。したがって、グラフの配置は適切に変更できることが望ましい。

3.2 アプローチ

GUIのみでデータ指定を行えるインタフェース

データの指定をリストボックスやラジオボタンの選択、テキストボックスへの入力といったGUIへの操作のみで行えるインタフェースにする。これにより、利用者は可視化プログラムの開発やデバッグといった作業をすることなく可視化を行うことができる。

データ取得とグラフ描画の分離

多様なフィルタリングに対応するため、クエリによってデータを取得する部分と、データを描画する部分を分離する。データを取得する部分では、基本的にデータベースに格納されているデータ全てを、任意のフィルタリングを通して取得できる。さらに、次節で述べるように、時相グラフや位置グラフで表せないデータもテーブルとして表示するため、本システムではデータベースに保存されているデータすべてを可視化できる。

テーブルのスキーマと描画可能グラフの対応付け

データベースに格納されたデータを用いて可視化を行う場合、描画可能なグラフの形式は、可視化対象のデータを含むテーブルの持つスキーマによって異なる。これを利用して3.1節で述べた描画可能なグラフ形式の自動判定を行う。表1に、定められたフィールドのスキーマがテーブル内にあるかないかによる場合分けの一例を示す。テーブル内に、timestamp型の時間を示すフィールドが存在する場合、データは時相グラフで描画可能である。また、"lat","lng"と命名された、数値で緯度経度情報を示すフィールドが存在する場合は、データを位置グラフとして描画可能である。これらが両方存在した場合には、どちらのグラ

表 1 フィールドによる描画可能グラフ
Table 1 Vizualizable Graph by Field Type.

	時間あり	時間なし
位置座標あり	時相グラフと位置グラフの描画が可能. どちらかを選択させる	位置グラフを利用して描画可能 どちらも描画できないため, テキストで表示
位置座標なし	時相グラフの描画が可能	

フも描画することができるため、利用者がどちらのグラフで描画するかを選択する。テーブルの持つフィールドに、時間と位置座標のどちらも含まれず、指定されたフィールドが数値でない場合は、グラフ化を行わずテーブルとしてデータを表示する。また、時相グラフや位置グラフ以外にも、そのフィールドのみで意味を持つ数値データであれば、円グラフや棒グラフ、レーダーチャート、散布図などで可視化が行えると考えられる。

協調動作させるスライドバーの設置

複数描画したグラフを協調動作させるためのトリガとして、画面内に時間軸を移動させるスライドバーを設置する。このスライドバーの変更値は各グラフに伝えられる。複数の同じ時間範囲のデータが描画されていた場合、スライドバーによりそれぞれのデータの注目点を同時に動かすことできる。これにより、利用者がそれぞれのデータの対応を容易に把握可能とする。

自由に配置可能なグラフ

3.1 で述べたように、解析内容によって、同時に描画したいグラフの数が異なる。さらに、グラフを描画する位置によって、解析しやすさは異なると考えられる。したがって、グラフは任意の数だけ表示可能にする。また、その表示箇所やサイズをマウスのドラッグアンドドロップ操作で自由に変更できる仕様になる。

4. 設計と実装

4.1 DUBIX の設計

本論文では 3.2 節で挙げたアプローチを実現する可視化システムとして、DUBIX を作成した。3.2 節で述べたアプローチを基に、本節では DUBIX の設計について述べる。本システムはインタフェース部、データ取得部、グラフ描画部の 3 つの構成要素から成る。以下にその説明を述べる。

インタフェース部 インタフェース部は描画パネルとスライドバーで構成されている。描画

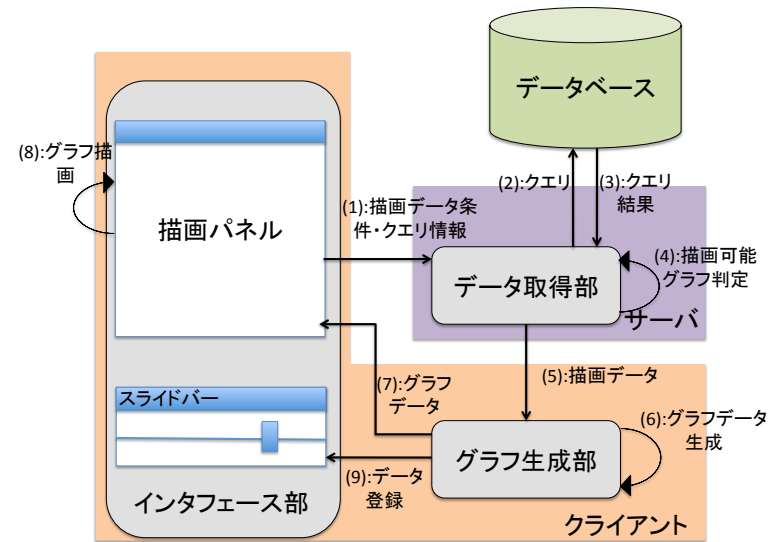


図 1 処理の流れ
Fig. 1 Process Flow.

パネルには、グラフが描画される。また、描画対象データの条件（データベース名、テーブル名、フィールド名、携帯端末 ID）を指定するためのリストボックスと、クエリを直接書くためのテキストボックス、描画の実行ボタンが設置されている。描画パネルは初期状態では 2 つ設置されており、マウスポインタを用いたドラッグアンドドロップによって、位置の移動が行える。さらに、描画パネルは任意の数追加可能である。スライドバーはブラウザの画面内にひとつだけ設置され、つまみを左右に動かすことができる。

データ取得部 データ取得部では、利用者が入力したデータの条件、あるいはクエリ情報を元にクエリを生成し、データベースから描画対象データを取得する。

グラフ生成部 グラフ生成部では、受け取ったデータを元にグラフ情報を生成する。

4.2 グラフ生成までの処理の流れ

4.1 節で述べた設計の、グラフ生成までの処理の流れを図 1 に示す。

(1) 利用者はグラフとして描画したいデータの条件を GUI で入力するか、直接 SQL 文を入力する。実行ボタンが押されると、描画パネルはその時点で利用者に選択されていた条件

か、入力されていたクエリ情報をデータ取得部に送る。

(2) データ取得部は描画パネルから受け取った条件か、SQL 文を元にクエリを生成し、データベースに送る。

(3) データベースからクエリ結果が返ってくると、データ取得部はグラフ形態の判別に必要なデータを再検索し、クエリ結果とまとめる。

(4) データ取得部は(3)のデータを基に、利用者が指定したデータがどの形態のグラフで描画可能かを自動で判定する。

(5) データ取得部は(3)で取得したクエリ結果と、(4)で判定したグラフ形態を、描画データとしてグラフ生成部に送る。

(6),(7) グラフ生成部ではデータ取得部から得た描画データを元に、グラフデータを生成し、描画パネルに送る。

(8) 受け取ったグラフデータを基に、描画パネルはグラフを描画する。

(9) また、グラフ生成部では 4.3 節で説明する各グラフの協調動作のために、グラフデータの生成と同時にそのグラフに対応したデータをスライドバーに登録する。

4.3 DUBIX の実装

DUBIX は Google Web Toolkit^{*1}を用いた Web アプリケーションとして、リストボックスとラジオボタンを利用した条件入力のみに対応するプロトタイプ実装を行った。また、描画パネルとスライドバー用のライブラリとして Ext GWT 2.2.0^{*2}を、線グラフを描画するためのライブラリとして Google Chart Tool API for GWT 1.1.0^{*3}を、地図上へデータを描画するためのライブラリとして Google Maps API for GWT 1.1.0^{*4}を利用した。参照するデータベースとしては、本研究室が坂本ら⁵⁾のフレームワークを用いて 1 年以上に渡って携帯端末から得られるライフログを蓄積したデータベースを用いる。このデータベースには GPS から得られた位置情報や、Wi-Fi 基地局情報などが時間をキーとして格納されている。

時相グラフの描画

図 2 に時相グラフを描画したときの様子を示す。選択されたテーブル内に timestamp 型の時間を表すフィールドが存在し、描画の対象データが数値であった場合、DUBIX はそのテーブルのデータが、時相グラフとして描画できると自動で判断し、時間指定のためのリス

トボックスを表示する。このとき、描画したいフィールドと、時間範囲を選択をして描画の実行ボタンを押すと、横軸に時間、縦軸に描画したいデータがセットされた線グラフが描画される。

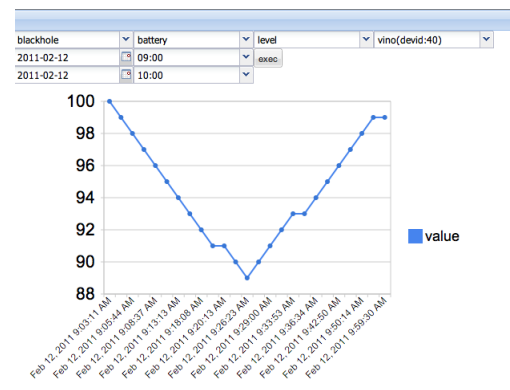


図 2 時相グラフを描画した様子

Fig. 2 Visualize Data as Time Graph.

位置グラフの描画

図 3 に位置グラフを描画したときの様子を示す。選択されたテーブル内に double 型で、"lat", "lng"と命名されたフィールドが存在する場合、DUBIX はそのテーブルのデータが、位置グラフとして描画できると判断する。このとき、入力欄を全て埋めて描画の実行ボタンを押すと、GoogleMaps を利用した地図と、その上に指定されたデータが表示される。

テーブルの描画

図 4 に Wi-Fi の essid をテーブルとして描画した様子を表す。選択されたテーブル内に timestamp 型のフィールドと、double 型で lat, lng と命名されたフィールドのどちらも存在しない、あるいは選択されたフィールドが数値でない場合、DUBIX は指定されたフィールドをグラフとして描画できないと判断する。この場合、DUBIX はデータをテーブルとして描画する。

描画可能なグラフの選択

描画可能なグラフ形態の選択の様子を図 5 に示す。時相グラフと位置グラフの両方を描画

*1 <http://code.google.com/intl/ja/webtoolkit/>

*2 <http://www.sencha.com/products/extgwt/>

*3 <http://code.google.com/p/gwt-google-apis/wiki/VisualizationGettingStarted>

*4 <http://code.google.com/p/gwt-google-apis/wiki/MapsGettingStarted>

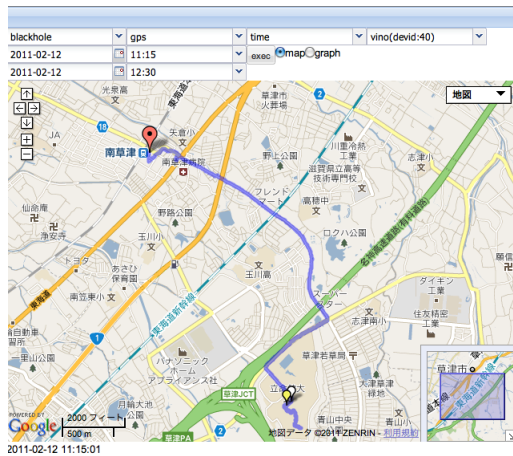


図3 位置グラフを描画した様子
Fig.3 Visualize Data on Map.

time	value
Apr 22, 2011 10:00:07 AM	Rits-Wireless
Apr 22, 2011 10:00:07 AM	045f47
Apr 22, 2011 10:00:10 AM	0024A5BA7A6E
Apr 22, 2011 10:00:10 AM	0024A5BA7A6E-1
Apr 22, 2011 10:00:10 AM	4CE67656B8B3
Apr 22, 2011 10:00:10 AM	GROUP

図4 テーブルを描画した様子
Fig.4 Visualize Data as Table.

できるとシステムが判断した場合、実行ボタンの近くに map, graph というラジオボタンが設置される。描画したいグラフをラジオボタンから選択し、描画の実行ボタンを押すと、選択されたグラフが生成される。

スライダーの移動による注目点の移動

DUBIX による複数グラフの協調動作の様子を図6に示す。この図では左側に時間による位置を、右側にその地点での速度を表示している。グラフが描画できた場合、ページ下部にあるスライダーを利用することで、グラフ上の注目点を動かすことができる。1つのグ

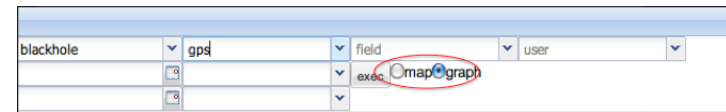
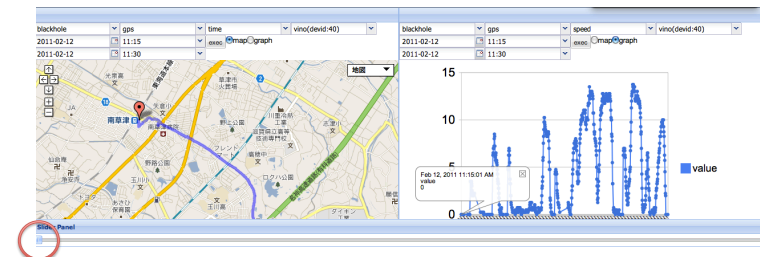


図5 描画可能なグラフの選択
Fig.5 Selection of Graph Type.



↓ スライダーを移動

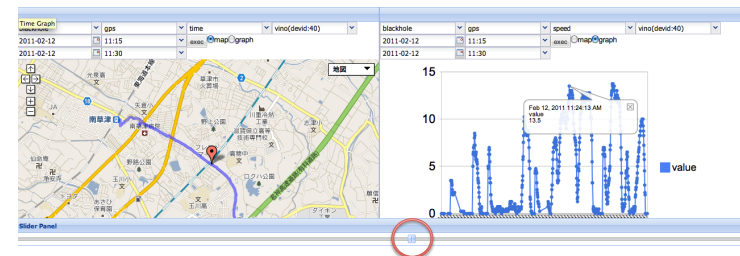


図6 スライダーの移動による協調動作
Fig.6 Collaborative Movement with Slider.

ラフだけでもこの機能を使うことができるが、複数の描画パネルにグラフが描画されていて、且つそのそれぞれに共通するデータがスライダーに登録されている場合、スライダーを動かすことによって複数のグラフの注目点を協調動作させることができる。

描画パネル及びスライダーの移動

描画パネルとスライダーは、それぞれパネルの上部をドラッグアンドドロップすること

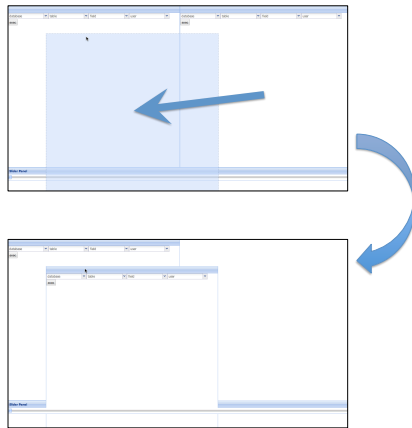


図 7 描画パネルの移動
Fig. 7 Panel Movement.

で、画面内に収まる範囲で移動することができる。図 7 にその様子を示す。

5. 評価と考察

5.1 定性評価

コードを書かずに少ない手順でグラフを描画

図 2 や図 3 で示したとおり、可視化したいデータはすべてブラウザ上のリストボックスから選択し、実行ボタンを押すだけでグラフ、あるいはテーブルとして描画できるようになった。つまり、新たなコードの記述を必要とせず、マウスポインタの操作のみでグラフを生成し、描画することができた。

多様なフィルタリングへの対応

データの取得先としてデータベースを利用し、時相グラフや位置グラフで描画を行うことができた。また、それらで描画できない場合もデータをテーブルとして表示することによって、データベース内全てのデータに対応した。しかし、フィルタリングに利用できるのは時間のみであるため、多様なフィルタリングへ対応したとは言い難い。

利用できるグラフ形式の自動判定

ユーザの描画したいデータが属するテーブルのスキーマによって、描画できるグラフの種類(時相グラフ、位置グラフ)をシステムが判定し、そのテーブルのデータに適したグラ

フを描画できるようになった。また、図 5 で示したとおり、描画できるグラフが複数種類あった場合、ユーザが選択できるような機構を設けた。これによって、ユーザは描画したいデータと描画したいグラフ、必要であれば時間を指定するだけでグラフを描画できるようになった。

複数のグラフの協調動作

図 6 で示したとおり、スライドバーの設置と、その操作による変更値を各描画パネルに通知することで、グラフ上の注目点を同時に、動的に移動させる機構を実現した。これによって、時間を軸としたデータの対応付けができ、ある位置における携帯端末利用者のスピードがすぐに分かるなど、データの解析が容易になったと考えられる。

グラフの描画位置指定

図 7 で示したとおり、描画パネルとスライドバーを画面内で自由に移動させることができた。これによって DUBIX の利用者は、複数のデータを比較・照合する作業に適した位置にグラフを配置することができる。

5.2 評価実験

表 2 にスライドバーの有用性、データ量に対するスライドバー操作の遅延時間、グラフの描画速度を確認するための評価実験で利用したマシンのスペックを示す。

5.2.1 スライドバーによる複数グラフの協調動作の有用性

本論文で提案してきた DUBIX の目的は、「マイニングプログラム開発者の負担軽減」である。このことから、マイニングプログラム開発者が本ツールを用いた場合、利用者が一番時間をかけて行うであろう、スライドバーによるデータの対応付けに関して、評価実験を行った。被験者は本研究室の男性 8 名である。

実験方法

まず被験者に対して DUBIX と注意事項の説明を行い、その後タスクを行ってもらった。このタスクはマイニングプログラム開発者が日頃行うであろう、2 つのデータの対応を参照するというものである。タスクの詳細を以下に示す。

表 2 クライアントの実験環境
Table 2 Client Specification.

CPU	Intel Core 2 Duo 2.53 GHz
OS	Mac OS X 10.6.7
メモリ	4 GB 1067 MHz DDR3
ブラウザ	Google Chrome13.0.761.0 dev

表 3 データセットによる分割
Table 3 Data Set.

	協調動作あり	協調動作なし
α	a ~ e	f ~ j
β	f ~ j	a ~ e

表 4 順序による分割
Table 4 Data Order.

	協調動作あり	協調動作なし
1	先	後
2	後	先

表 5 平均タスク終了時間
Table 5 Result of Experiment.

	データセット	協調動作あり (秒)	協調動作なし (秒)
A	α 1	10.01	17.00
B	α 1	10.03	27.13
C	α 2	9.23	17.61
D	α 2	8.71	23.11
E	β 1	12.17	29.34
F	β 1	10.50	21.58
G	β 2	8.68	24.14
H	β 2	9.59	25.89

- (1) GPS から一定時間内に得られた携帯端末の速度情報と、GPS の精度情報を同時に DUBIX と、協調動作をしない DUBIX で参照してもらう。
- (2) その時間内で、精度の値が一番高かった時刻に対する速度を口頭で答えてもらう。
1 人につき DUBIX と DUBIX(協調動作なし) で 5 回ずつ、計 10 回行ってもらう、タスク完了までの時間を測定した。タスクに使うデータは a~j までの 10 種類用意し、結果の偏りを防ぐためにその組み合わせ (表 3) と行う順序に (表 4) よってタスクを α 1, α 2, β 1, β 2 の 4 種類用意した。

実験結果

被験者 A~H がタスクに用いたデータセットと、平均タスク終了時間を表 5 に示す。

被験者 8 人の内全員が、協調動作なしの場合の DUBIX を利用したときよりも協調動作ありの場合の DUBIX を利用したときの方が、タスクを早く終わらせる事ができていた。また、この被験者の平均タスク終了時間を t 検定で分析した。「協調動作ありの DUBIX は協調動作なしの DUBIX と比較して差がない」という帰無仮説 H を立て、検定を行った結果、 $t = -8.37$ となった。ここで自由度は 14 であり、有意水準 1% が 2.977 であることから、帰無仮説は棄却される。つまり、協調動作の有無で有意差があったと確認できた。

5.2.2 スライダーを操作したときの遅延

ひとつのグラフ、あるいは複数のグラフの注目点をスライダー操作によって移動させて

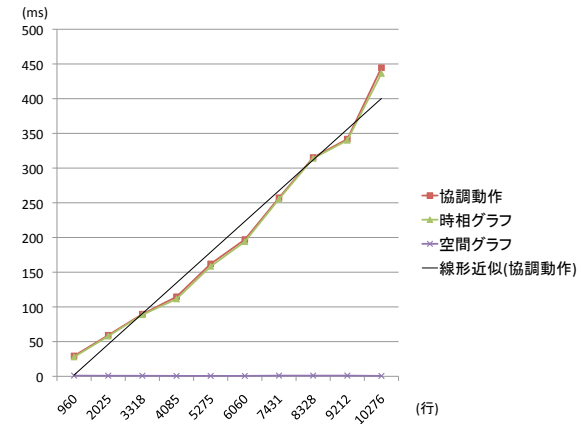


図 8 データ量に対するスライダーの遅延時間
Fig. 8 Slider Delay.

分析を行う場合、その動作の快適さが求められる。ここでスライダーの遅延時間は、データ量に対してスライダーが注目点から別に注目点に移動した時の描画パネルの再描画時間であると見なすことができる。そこで、Google Chrome の拡張機能である Speed Tracer^{*1}を用いて、データ量ごとのスライダーの遅延時間を計測した。その結果を図 8 に示す。

位置グラフに関しては、今回描画した最大数のデータ (10276 行) でも再描画にかかる時間は 1ms 前後で問題なくスライダーを動作させることができた。しかし、時相グラフでは、5000 行程度のデータを描画した場合は、再描画に 160ms 程度かかり、自分で意図した部分に注目点を合わせる事が困難であった。更に、1 万行程度のデータを描画した場合は、436ms 程度かかり、動かすこと自体が困難であった。この結果より、現在の DUBIX では時相グラフを描画した場合にデータ量によってかなり遅延が発生してしまうことが確認できた。この原因としては、線グラフに用いている API が、大量のデータを描画した場合に注目点を変える動作を高速に行えていないことが考えられる。この問題を解決するために、大量のデータに対しても高速に処理を行える API を採用するか、作成する必要がある。

5.2.3 グラフの描画速度

今回実装した DUBIX は、扱うデータの量によって、動作への影響が異なると考えられ

*1 <http://code.google.com/intl/ja/webtoolkit/speedtracer/>

表 6 データ量に対するグラフの描画時間
Table 6 Visualizing Time by Quantity of Data.

データ量 (行)	位置グラフ (秒)	時相グラフ (秒)
960	約 0.56	約 0.48
4885	約 0.75	約 0.97
10276	約 1.51	約 2.68
30082	約 3.49	約 19.62

る。このことから、5.2.2 節と同じく Speed Tracer を用いた描画速度の評価を行った。表 6 に、データ量に対する位置グラフおよび時相グラフの描画にかかった時間を提示する。

この結果より、1 万行程程度のデータであれば、時相グラフ、位置グラフ共に数秒で描画を行えることがわかった。しかし、3 万行のデータになると、時相グラフでは描画に 20 秒近く時間がかかることを確認した。この原因としては、5.2.2 節と同様に、線グラフを作成するための API がこのデータ量に対して処理を高速に行えていないのが原因と考えられる。

5.3 考 察

評価を行ったことにより確認した考察を述べる。

5.3.1 データの指定

現状の DUBIX では、ユーザが利用できるのはリストボックスとラジオボタンによる条件指定のみで、描画対象のデータを時間範囲でしかフィルタできない。ライフログの解析作業やマイニングプログラムのデバッグ作業では、時間以外の条件、例えば値が一定以上のデータを参照することなども十分に考えられる。したがってもっと多様な条件を指定できる必要がある。

グラフ描画 API への依存

4.3 節で述べたように、現在グラフの作成には外部の API を用いている。このような API は種類も豊富で、多機能であるが、実際のソースコードを扱おうとすると細かい部分で融通が効かない場合が多い。したがって、外部の API だけに頼らず、ライフログに特化したグラフ作成用 API が必要だと考えられる。

マイニング結果を描画するモジュールの不足

DUBIX はライフログもマイニング結果もグラフとして表示可能である。しかし、現状描画を行えるのは時相グラフと位置グラフ、テーブルのみであるため、効果的にライフログの解析やマイニングプログラムのデバッグを行えるとは言い難い。したがって、各種データをより効果的に描画できる描画モジュールの考案と開発が必要であると考えられる。

6. 結 論

本論文では、マイニングプログラム開発における、データ解析やプログラムのデバッグなどといった作業の負担を減らすことを目的とし、ライフログ可視化システムである DUBIX を提案し、その実装と評価を行った。複数のグラフを同一画面内に設置し、それらをスライドバーを用いて協調動作させ、動的に注目点を移動させるといった描画を行うことで、静的なグラフを並べるシステムよりも、データの対応を効率よく参照できる可視化システムを実現することができた。

しかし、現在 DUBIX は GUI を経由したクエリしかサポートしておらず、ユーザの直接入力するクエリには対応していない。したがって、今後はこれに対応する必要がある。また、5.3 節で述べたようにマイニング結果に対しても、外部の API に頼らず、データの参照を効果的に行えるような可視化モジュールの考案・開発を行い、システムを充実させていきたい。

参 考 文 献

- 1) 小塚宜秀. ユビキタスネットワーク技術の研究開発 - ケータイ de ライフログ -. Technical Report 3, KDDI 研究所, 2008
- 2) 森岡英之, 岩田直樹, 新津善弘. 複数センサを用いたコンテキスト推定法. 電子情報通信学会東京支部学生会研究発表会, 2009.
- 3) 太田健吾, 向井政貴, 新井イスマイル, 西尾信彦. Wi-Fi 基地局の観測履歴を利用したユーザの状態遷移解析. マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO2010), 2010.
- 4) 堀之内武, 西澤誠也, 渡辺知恵美, 森川靖大, 神代剛, 石渡正樹, 林祥介, 塩谷雅人. 地球流体データベース・解析・可視化のための新しいサーバ兼デスクトップツール Gfdnavi の開発. 電子情報通信学会 第 18 回データ工学ワークショップ論文集, 2007.
- 5) 坂本憲昭, 坂本一樹, 名生貴昭, 市川昌宏, 新井イスマイル, 西尾信彦. 同期シナリオを用いてセンシング携帯端末と協調連携するアプリケーションフレームワークの提案. マルチメディア, 分散, 協調とモバイルシンポジウム (DICOMO2010), 2010.