

# わかるらんど: IoT 時代の情報共有視覚化

山田 尚昭<sup>1</sup> 増井 俊之<sup>2</sup>

**概要:** 全世界のセンサ情報やユーザの気分などを一覧表示したり投稿したりできるシステム「わかるらんど」を提案する。ニュースや天気予報のようなリアルタイム情報を並べて一覧表示する「情報ダッシュボード」の利用が広まっているが、利用できる情報の種類は限られており、ユーザが情報を投稿して共有することはできない。わかるらんどは、単純で強力な Web 上の情報共有システム「WebLinda」上に構築された汎用的な情報共有/視覚化システムであり、ユーザの気分を表明したり、チャット文字列を投稿したり、センサ情報や Web 上の情報を表示したり、ネット上のあらゆる情報を投稿/共有して一覧表示することができる。わかるらんどの利用により、情報ダッシュボードと SNS やチャットシステムを簡単に統合的に利用することができる。本論文では、わかるらんどの思想及び利用経験について述べ、応用について考察する。

## WakaruLand: An information visualization system for the IoT age

NAOAKI YAMADA<sup>1</sup> TOSHIYUKI MASUI<sup>2</sup>

**Abstract:** We propose a new information visualization system “WakaruLand” that can show various real-world information including sensing data, information on the Web, people’s activities and feelings, etc. Although *information dashboard* systems are getting popular, they can display limited sets of information provided by the services, and people cannot freely add data channels or post arbitrary information related to them. Using WakaruLand, people can easily create new data sources and post their activities and feelings using the WebLinda system running on a Web server, using a simple and powerful communication protocol based on Linda,

We introduce how we can use WakaruLand for various applications including information dashboard, visualization of users’ status, etc.

## 修正とか必要なところ

- ・「はじめに」でわかるらんどの思想を述べる
- ・消極性コンピューティング [6] について関連システムか何かとして述べる
- ・linda-server じゃなくて WebLinda にすればどうか
- ・Ubicomp 資料 [3] に WebLinda の話は書いてたかも
- ・昔の WISS チャットの問題点と、わかるらんどでの解決期待を書いてください (特定の人ばかり投稿していることを示す)
- ・N 人の参加者の投稿が 90% になってた、みたいな
- ・Linda の参考文献も [1]

## 1. はじめに

ワイヤレスネットワークや小型計算機の普及による IoT 社会が到来しつつある現在、人々は大量のリアルタイム情報や通知やメッセージなどに圧倒されている。多くの情報を人間が理解しやすくするため、以下のような視覚化手法が利用されている。

### 情報ダッシュボード

情報ダッシュボード [2] は、複数のリアルタイム情報をタイル状に並べて表示することによって多くの情報をわかりやすく視覚化するシステムである。たとえば Windows のスタート画面 (図 1) の情報ダッシュボードには天気予報や株価のようなリアルタイム情報を表示可能である。

<sup>1</sup> 慶應義塾大学大学院 政策・メディア研究科

<sup>2</sup> 慶應義塾大学 環境情報学部

## スタンプ

リアルタイムに流れていくタイムライン表示の中で情報を目立たせたいとき、最近「スタンプ」と呼ばれるピクトグラムが利用されることが多くなってきた(図2)。

スタンプはテキストで記述するのが難しい表現や感情を伝えたり、テキストを考えて入力するよりも速くて簡単であったりすることから、近年LINEやFacebookメッセンジャー、オンラインゲームなどで広く利用されている。



図2 LINEのスタンプの例

スタンプ的な表現を投稿可能な情報ダッシュボードを用意し、その上でネット上の様々な情報やユーザの気分や感情を表示すれば、現在の世界や人々の状況が一目了然になる。実世界の状態や人間の状況を情報ダッシュボードにわかりやすく表示し、かつ誰もが簡単に気分などをスタンプのように投稿して共有できる「わかるらんど」システムを構築した。

## 2. わかるらんど

本章では、我々の開発した『わかるらんど』の機能と使用方法について述べる。

### 2.1 ユーザインターフェース

『わかるらんど』のユーザインターフェースは「ダッシュボード」と「投稿画面」の2つからなる。図3はダッシュボードのスクリーンショットである。インフォメーションダッシュボードのようにセンサのデータを表示したり、ユーザがリアクションとしてテキストやスタンプを投稿したものを見ながらリアルタイムに1つの画面に表示することができる。



図1 Microsoft Azureの情報ダッシュボード

る。センサのデータなどは自動的に更新され、ユーザのリアクションは図4の投稿画面でユーザ名を入力し、スタンプを一覧から選んで投稿することで自分のアイコンの上にオーバーレイ表示される。ダッシュボードと投稿画面は画面右上のボタンで切り替えることができる。また、投稿するスタンプは投稿画面でユーザが自由に追加/削除することができる。



図3 『わかるらんど』のダッシュボード



図4 『わかるらんど』の投稿画面

### 2.2 表示するユーザとデータの指定

『わかるらんど』のダッシュボードに表示されるセルは、ユーザのテキストや画像のスタンプを表示する「リアクション」セルと、センサやWebの情報を表示する「データ」セルの2種類がある(図5)。



図5 リアクション(左)とデータ(右)

ダッシュボードに表示するユーザのリアクションの指定は、<https://wakarulando.com/@napo0703,@masui,>

@shokai, @dorayaki0 のように URL の末尾にカンマ区切りで Twitter のユーザ名を「@」を付けて書くことで行う。この場合は図 6 のように @napo0703, @masui, @shokai, @dorayaki0 の 4 人のユーザのセルがダッシュボードに表示される。



図 6 4 人のユーザを表示

<https://wakaruland.com/weather,temperature,wind> のように「@」を付けなかった場合は図 7 のように weather,temperature,wind の 3 つがデータのセルとして表示される。

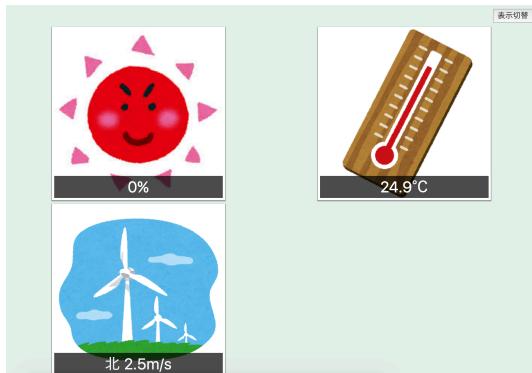


図 7 3 つのデータを表示

また、<https://wakaruland.com/@napo0703,weather,@masui,wind,@shokai> のようにリアクションとデータを合わせて表示させることも可能である（図 8）。

### 2.3 スタンプの作成

ユーザが投稿画面でリアクションとして投稿するスタンプには、「画像」スタンプと「テキスト」スタンプの 2 種類がある（図 9）。ユーザは投稿画面のテキストボックスに文字列を入力し追加ボタンを押すことで新たにスタンプを作成して追加することができる。

画像スタンプは、テキストボックスに <http://masuilab.org/image.jpg> のような Web にある画像の URL を入力し追加ボタンを押すことで、URL の画像をスタンプとして一覧に追加することができる。ローカルにある画像ファ



図 8 ユーザとデータを表示

イルをスタンプとして追加したい場合は、Gyazo などのアプリケーションを使って Web にアップロードし URL を得ることで追加が可能である。

テキストスタンプは、テキストボックスに URL でない文字列を入力し追加ボタンを押すことで作成できる。図 10 左は「わからん」とテキストボックスに入力してスタンプを作成したものである。この「わからん」は 1 行に表示されているが、「わからん」と改行を入れたい場所に半角スペースを入力することで、図 10 右のように 2 行で表示される。

スタンプの削除は、スタンプにマウスオーバーすると右上に表示される「×」ボタンを押すことで削除できる。



図 9 テキストスタンプ（左）と画像スタンプ（右）



図 10 「わからん」スタンプ（左）と「わからん」スタンプ（右）

### 2.4 利用例

図 11 は会議や発表の場での利用例である。これは誰かが何か変なことを言ったのに対してユーザが反応している様子である。

図 12 はダッシュボードとしての利用例で、各種のセンサの値や Web の情報等を表示している。データはリアルタイムに最新のものに更新されていく。



図 11 会議や発表の場での利用例

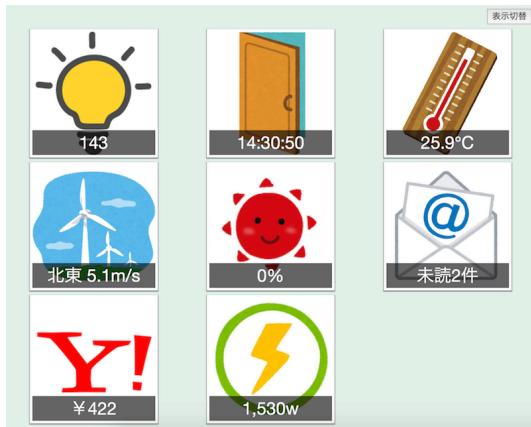


図 12 ダッシュボードとしての利用例

### 3. 実装

本章では『わかるらんど』の実装について述べる。

クライアントは HTML/CSS/JavaScript で実装しており、通常のブラウザ上で動作する Web アプリケーションとして動作する。

サーバは並列計算プリミティブ Linda を Web サーバ上に実装した WebLinda<sup>\*1</sup>を用いて実装している。

#### 3.1 Linda

Linda[1] は、複数のプロセスで共有される空間を用いてプロセス間通信やデータ共有をサポートする分散並列処理を行うためのモデルである。プロセスが共有する空間はタプル空間 (Tuple Space) と呼ばれ、タプル空間内のデータ (Tuple) を使って通信やデータ共有を行う (図??)。このように Linda のモデルはきわめて単純であるが、各クライアントやデバイス間で直接送信をする処理を記述する必要がなく、柔軟で強力なプロセス間通信を容易に記述することができる。

<sup>\*1</sup> <https://github.com/node-linda/linda>

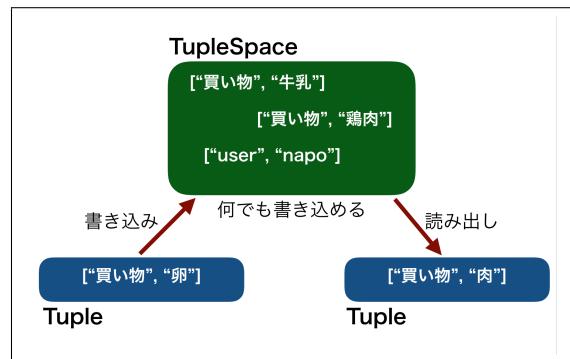


図 13 タプルスペースのイメージ

#### 3.2 WebLinda

WebLinda は、橋本翔<sup>\*2</sup>氏が開発したオープンソースソフトウェアで、Node.js<sup>\*3</sup>の WebSocket ライブライアリ Socket.IO<sup>\*4</sup>上に実装された Linda システムである。通常の Web サーバ上に実装されているため、HTTP 通信ができれば様々な環境やプログラミング言語で WebLinda を使うことができる。WebLinda は、write, read, take, watch の 4 つの基本操作によってプロセス間通信を行う。

##### write

新しいデータオブジェクト（タプル）を生成し共有空間（タプルスペース）に書き込む。

##### read

指定した形式に部分一致するタプルがタプルスペースにあるかどうか調べて 1 つ読み出す。一致するものが無い場合は一致するタプルが書き込まれるまで待つ。

##### take

read しつつ、読み出したタプルをタプルスペースから削除する。

##### watch

タプルスペースを監視し、一致するタプルが write された瞬間に読み出す。

#### 3.3 『わかるらんど』での WebLinda 実装

『わかるらんど』での WebLinda 実装について述べる。ユーザのリアクションを投稿/表示する際には、

```
{
  type: "reaction",
  from: "@napo0703",
  display: "60",
  value: "なるほど"
}
```

というようなタプルをやりとりしている。

<sup>\*2</sup> <http://shokai.org>

<sup>\*3</sup> <https://nodejs.org>

<sup>\*4</sup> <http://socket.io>

```

センサや Web のデータを投稿/表示する際には、
{
  type: "data",
  from: "明るさ",
  display: "60",
  value: "500",
  background: "http://masuilab.org/image.jpg"
}

```

というようなタプルをやりとりしている。

#### type

ユーザのリアクションの場合は reaction、データの場合は data を値にする。

#### from

投稿元を表す値。

#### display

リアクションやデータの表示時間。単位は秒。リアクションやデータが表示されてからこの秒数が経過すると、自動的に取り下げられ表示されなくなる。20～86400 の間で指定が可能。

#### value

リアクションの場合、この値が Web にある画像ファイルの URL だった時は、その画像を投稿者のセルにオーバーレイ表示する。URL でない文字列の場合は、その文字列を投稿者のセルにオーバーレイ表示する。データの場合、この値がそのままセルの下部に表示される。

#### background

(データのみ) データセルの背景画像の URL。そのデータが何を表すものなのか分かる画像を表示する。

以下は、ボタンを押した際に『わかるらんど』にユーザ「@napo0703」として「なるほど」という「リアクション」を「20秒」表示するタプルを書き込む JavaScript プログラムの例である。

```

// Linda に接続
const url = "http://linda.masuilab.org";
const socket = SocketIO(url);
const linda = new Linda().connect(socket);
const ts = linda.tuplespace("masuilab");

// ボタンを押したらタプルスペースにデータ書き込み
const button = $("button").click(function () {
  ts.write({
    type: "reaction",
    from: "@napo0703",
    display: "20",
    value: "http://masuilab.org/image.jpg",
  });
})

```

```

display: "20",
value: "http://masuilab.org/image.jpg",
});
})];

```

以下は、先程のプログラムで書き込まれるタプルと合致する {from: "@napo0703", type: "reaction"} を含むタプルが書き込まれた場合にリアクションを表示する JavaScript プログラムの例である。

```

// Linda に接続
const url = "http://linda.masuilab.org";
const socket = SocketIO(url);
const linda = new Linda().connect(socket);
const ts = linda.tuplespace("masuilab");

```

```

// 合致するタプルが書き込まれたら画像を変える
ts.watch({
  from: "@napo0703",
  type: "reaction"
}, function (tuple) {
  $("img").attr("src", tuple.data.value);
});

```

以下は現在の江ノ島の風向と風速をタプルスペースに書き込む Ruby プログラムである。Nokogiri というライブラリで Web ページから必要な情報をスクレイピングしてタプルスペースに書き込んでいる。

```

url = 'http://linda.masuilab.org'
linda = Linda::SocketIO::Client.connect url
ts = linda.tuplespace('masuilab')

ts.on :connect do
  // Nokogiri でスクレイピング
  web = 'http://www.s-n-p.jp/kishou.htm'
  doc = Nokogiri::HTML(open(web))
  // 必要な情報を取得
  tds = doc.xpath("//td")
  as = tds[16].xpath("./b").text.sub(/[^\\d\\.]*$/,'')
  ad = tds[19].xpath("./b").text
  // タプルスペースに書き込み
  ts.write(
    where: 'enoshima',
    type: 'data',
    name: 'wind',
    value: ad + as.to_f + 'm/s')
end

```

## 4. 議論

### 4.1 『わかるらんど』の思想

講義やコンファレンスではタイムライン表示のテキストチャットが利用されることが多いが、以下のような問題があると考えている。

- 同時に多数が投稿するとすぐに流れていってしまう
- 投稿数の多い人ばかりが目立ってしまう
- 投稿しない人は全く投稿しない

チャットに限らず実際の議論でも特定の人だけが沢山発言し、発言しない人は全く発言しない状況はよくあることだ。

「On Air Forum」は、日本ソフトウェア科学会主催のWISSで利用されているコミュニケーションシステムである。「On Air Forum」のWISS2009の実証実験[7]では、全参加者の半分弱しかログインして1回以上発言していない。WISS2015では、252アカウントが1回以上発言し総発言数は2,948回であったが、発言数上位20%の50アカウントによる発言が総発言数の78.1%にあたる2,305回を占めていた(図14)。また、発言数が10回未満のアカウントは190アカウントで、これは全アカウントの75.3%にあたる。

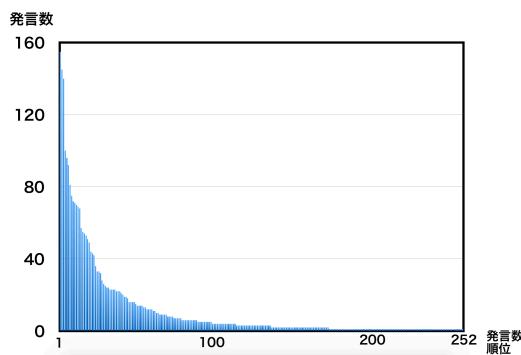


図 14 WISS2016 のチャットにおけるアカウント毎の発言数

『わかるらんど』はユーザの表示領域が均等に決まっているため、タイムライン表示のように投稿数が多い人ばかりが目立つということがない。また、長いテキストを入力すると表示される文字が小さくなるので、150人程度で利用すると長い文字は小さすぎて読めない。必然的にユーザは図15のように短文を入力することを強いられる。『わかるらんど』では長文の高度な発言は期待しておらず、「なるほど」「わからん」「笑」などといった相槌のようなものを視覚化してひと目で把握できるようになることを期待している。学生、先生、企業に所属する人等様々なバックグラウンドの人が入り混じった状況で「下手な発言ができない」「気の利いたことを言わなければならない」という投稿を躊躇させる要素を限りなく減らし、本当は議論に参加し

たいけど声が出ない/手を上げる勇気がない人でも「なるほど」「わかる」などを『わかるらんど』に投稿することで「参加」することができる。テキストで記述すると長くなってしまう内容も画像スタンプを投稿することで分かつてもらうことができると考える。また、長いテキストを投稿するには適していないので『わかるらんど』を使って議論することは難しいが、多くの会議やコンファレンスでは発表後に議論の時間が設けられているため議論はその時に行えばよい。



図 15 150 人で『わかるらんど』を使用したイメージ

### 4.2 研究室においての利用経験

我々の研究室では、研究室内のディスプレイに研究室メンバー全員のリアクションと部屋の明るさ、温度、最後に研究室のドアが開いた時間を表示した『わかるらんど』を常に表示して、約6ヶ月の間利用してきた。ミーティングの時間には積極的に『わかるらんど』を利用した。普段発言の少ない人でも何らかの反応を表明したり、誰かが面白いことを言ったときに「笑」というリアクションが並ぶと面白かった。また発表者としても、聴衆が自分の発表を聞いてくれているかどうかわからないときに『わかるらんど』にリアクションを投稿してもらうことで、聴衆がみんなPCの画面を見ていても何らかの投稿があれば話を聞いているということがわかるようになった。自分が研究室にいないときでもリアクションを投稿して、研究室でも自宅でも研究室メンバーの気分や研究室のセンサ情報が見られるようになった。

研究室で利用しているチャットでチャットボットに話しかけることで、研究室の

- ドアの鍵を開ける
- 照明を点ける／消す
- 明るさを知る
- 気温を知る

ことができるが、『わかるらんど』と組み合わせることで非常に便利に使うことができた。

## 5. 関連研究

ダッシュボードは多くの製品やサービスが存在する。研究としては、2つに区切られたレイアウトのダッシュボードのセルの配置を支援するもの[4]や、ある課題の解決のためにどのような情報をダッシュボードに表示するべきか[5]などが議論されている。ダッシュボードに人間の感情や現在の状況を表示するといった試みは今までに行われていないと思われる。

## 6. 結論

人や環境の状態がリアルタイムにわかる視覚化システム「わかるらんど」を提案した。非常に汎用で拡張性も高く、様々な場面での情報共有に利用されることが期待でき、IoT時代の情報視覚化として広く利用されるとともに、長い間続けられてきたテキストベースのコミュニケーションに終止符を打ち、ピクトグラムベースのコミュニケーションの先駆け的存在となるだろう。

## 参考文献

- [1] Carriero, N. and Gelernter, D.: Linda in Context, *Commun. ACM*, Vol. 32, No. 4, pp. 444–458 (1989).
- [2] Few, S.: *Information Dashboard Design: Displaying data for at-a-glance monitoring*, Analytics Pr; (2013).
- [3] Hashimoto, S. and Masui, T.: The Furniture of Ubiquitous Computing, *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, UbiComp '13 Adjunct, New York, NY, USA, ACM, pp. 845–852 (2013).
- [4] Hertzog, P.: Binary Space Partitioning Layouts To Help Build Better Information Dashboards, *Proceedings of the 20th International Conference on Intelligent User Interfaces*, IUI '15, New York, NY, USA, ACM, pp. 138–147 (2015).
- [5] Jones, S. L.: Exploring Correlational Information in Aggregated Quantified Self Data Dashboards, *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*, UbiComp/ISWC'15 Adjunct, New York, NY, USA, ACM, pp. 1075–1080 (2015).
- [6] 栗原一貴, 西田健志, 濱崎雅弘, 築瀬洋平, 渡邊恵太: 消極性デザイン宣言 – 消極的な人よ、声を上げよ。……いや、上げなくてよい。, ビー・エヌ・エヌ新社 (2016).
- [7] 西田健志, 栗原一貴, 後藤真孝: On-Air Forum: リアルタイムコンテンツ視聴中のコミュニケーション支援システムの設計とその実証実験, コンピュータソフトウェア, Vol. 28, No. 2, pp. 2 183–2 192 (2011).