

わかるらんど: IoT時代の共有情報視覚化

山田 尚昭¹ 増井 俊之²

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 YAMADA¹ TOSHIYUKI MASUI²

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 these days, they can display limited sets of information provided by the services, and people cannot freely add data sources 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 show how we can use WakaruLand for various applications including information dashboard, visualization of users’ status, etc.

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社会が到来しつある現在、人々は大量のリアルタイム情報や通知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に、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。



図1 Windows のスタート画面

スタンプ

リアルタイムに流れしていくタイムライン表示の中で情報

¹ 慶應義塾大学大学院 政策・メディア研究科

² 慶應義塾大学 環境情報学部

を目立たせたいとき、近年「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情報やユーザの感情/気分などを表示すれば、現在の世界や人々の状況を一目了然に理解する(わかる)ことが可能になるであろう。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つからなり、画面右上のボタンで切り替えることができる。図3はダッシュボードのスクリーンショットである。ユーザが投稿したテキストやスタンプや、各種のセンサのデータなどがリアルタイムに1つの画面に表示されている。センサのデータなどは自動的に更新され、ユーザのアクションは図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。ユーザの投稿には表示時間が設定されており、指定時間が経過すると自動的に投稿が取り下げられるためいつまでも古い投稿が表示され続けてしまうということはない。表示時間はユーザがスタンプをクリックする時間の長さで設定される。

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、



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

ユーザのテキストや画像のスタンプを表示する「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 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アクションの指定は、[@masui, @shokai, @dorayaki0](https://wakaruland.com/@napo0703)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。

<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。

また<https://wakaruland.com/@napo0703,weather,@masui,wind,@shokai>のように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)。



図 6 4人のユーザを表示

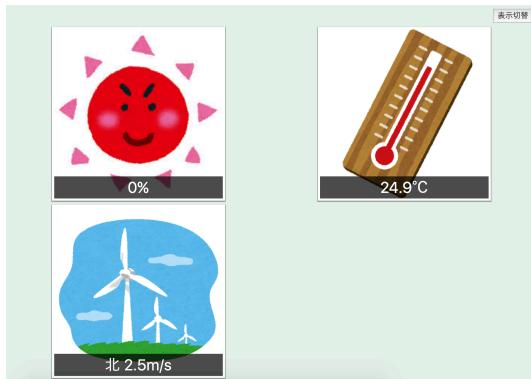


図 7 3つのデータを表示

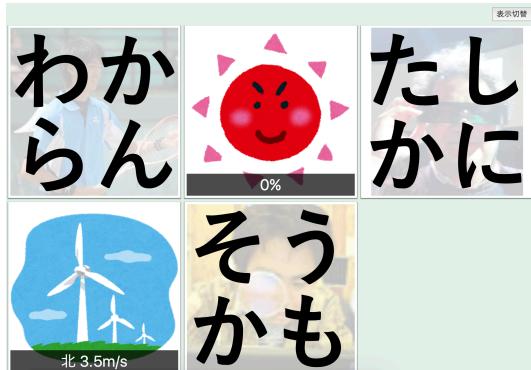


図 8 ユーザとデータを表示

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、「テキスト」スタンプと「画像」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て追加ボタンを押すことで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イルをスタンプとして追加したい場合は、Gyazo^{*1}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空白文字を入力することで、図10右のように2行で表示される。



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

^{*1} <https://gyazo.com>

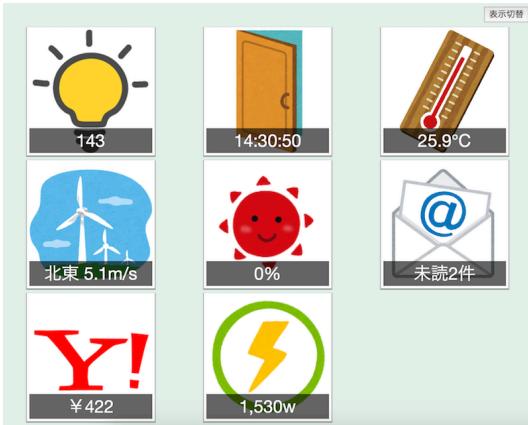


図 12 ダッシュボードとしての利用例

3. 実装

『わかるらんど』のクライアントは HTML/CSS-/JavaScript で実装されており、通常のブラウザ上の Web アプリケーションとして動作する。サーバは、並列計算 プリミティブである *Linda*[1] を Web サーバ上に実装した *WebLinda*[3]^{*2} を用いて実装している。

3.1 Linda

Linda 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 13）。*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。

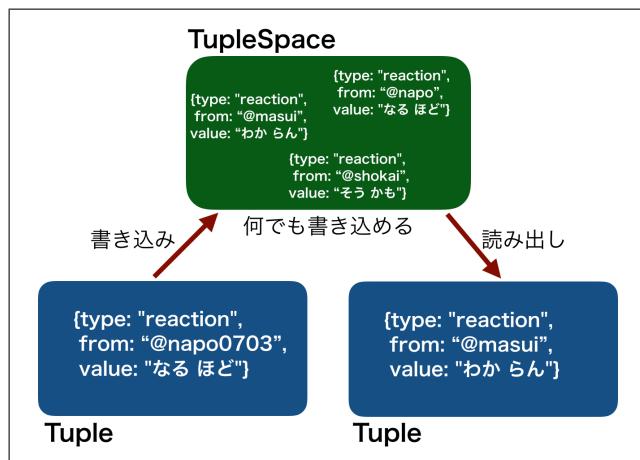


図 13 タプル空間を利用したプロセス間通信

3.2 WebLinda

WebLinda は、橋本翔氏が開発したオープンソースソフトウェアで、Node.js^{*3} の WebSocket ライブライアリ「Socket.IO^{*4}」で実装されている。*WebLinda* は通常の Web サーバ上に実装されているため、HTTP 通信をサポートする様々な環境やプログラミング言語で利用可能である。それゆえ、Arduino^{*5} や Raspberry Pi^{*6} などといった各種の IoT 機器との連携も容易である。

WebLinda は、write, read, take, watch という 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（タプル空間）に書き込む。

read

タプル空間からタプルを読み出す。

take

read しつつ、読み出したタプルをタプル空間から削除する。

watch

タプル空間を監視し、一致するタプルが write された瞬間に読み出す。

3.3 WebLinda による『わかるらんど』の実装

『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、センサと『わかるらんど』の間で

```

1 {
2   type: "reaction",
3   from: "@napo0703",
4   display: "60",
5   value: "なるほど"
6 }
```

というようなタプルをやりとりしている。

センサや Web のデータを投稿/表示する際には、

```

1 {
2   type: "data",
3   from: "明るさ",
4   display: "60",
5   value: "500",
6   background: "http://masuilab.org/image.jpg"
7 }
```

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。

^{*3} <https://nodejs.org>

^{*4} <http://socket.io>

^{*5} <https://www.arduino.cc>

^{*6} <https://www.raspberrypi.org>

^{*2} <https://github.com/node-linda/linda>

```

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 .

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 .

```

1 // Linda に接続
2 const url = "http://linda.masuilab.org";
3 const socket = SocketIO(url);
4 const linda = new Linda().connect(socket);
5 const ts = linda.tuplespace("masuilab");
6
7 // ボタンを押したらタプル空間にデータ書き込み
8 const button = $("button").click(function () {
9   ts.write({
10     type: "reaction",
11     from: "@napo0703",
12     display: "20",
13     value: "http://masuilab.org/image.jpg",
14   });
15 });

```

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 .

```

1 // Linda に接続
2 const url = "http://linda.masuilab.org";
3 const socket = SocketIO(url);
4 const linda = new Linda().connect(socket);
5 const ts = linda.tuplespace("masuilab");
6
7 // 合致するタプルが書き込まれたら画像を変える
8 ts.watch({
9   from: "@napo0703",
10   type: "reaction"
11 }, function (tuple) {
12   $("img").attr("src", tuple.data.value);
13 });

```

以下は現在の江ノ島の風向と風速をタプル空間に書き込む Ruby プログラムである . Nokogiri^{*7} 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空間に書き込んでいる .

```

1 url = 'http://linda.masuilab.org'
2 linda = Linda::SocketIO::Client.connect url
3 ts = linda.tuplespace('masuilab')
4
5 ts.on :connect do
6   // Nokogiri でスクレイピング
7   web = 'http://www.s-n-p.jp/kishou.htm'
8   doc = Nokogiri::HTML(open(web))
9   // 必要な情報を取得
10  tds = doc.xpath('//td')
11  as = tds[16].xpath('.//b').text.sub(/[^\\d\\.\\.]*$/,'')
12  ad = tds[19].xpath('.//b').text
13  // タプル空間に書き込み
14  ts.write(
15    where: 'enoshima',
16    type: 'data',
17    name: 'wind',
18    value: ad + as.to_f + ' m/s')
19 end

```

4. 議論

4.1 『わかるらんど』のねらい

近年 , 学会などでタイムライン表示のテキストチャットが利用される機会が増えている [9] . 学会チャットシステムを利用すると , 発表中に参加者が意見交換したり疑問を表明したりできるといった利点があるが , 以下のような問題も存在する .

- 多数の人間が同時に投稿すると投稿内容がすぐに見えなくなってしまう
- 投稿の多いアクティブな人ばかりが目立ってしまい , 消極的な参加者は議論に参加しにくい

一般に , 会議などで特定の人だけが沢山発言するのはよくあることであるが , 誰もが気軽に意見を表明できる環境を構築できれば有意義であろう .

日本ソフトウェア科学会主催の WISS (Workshop on Interactive Systems and Software) コンファレンス^{*8} では , 学会が提供するチャットシステムに参加者がログインして議論するのが恒例になっており [10] , 2009 年以降の WISS コンファレンスでは「On Air Forum」[7] という学会チャットシステムが利用されている . しかし , WISS2009 の実証実験によれば , 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かった . また 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) . 発言

^{*7} <http://www.nokogiri.org>

^{*8} <http://wiss.org/>

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 15 のようにアカウントと発言数は幕分布になっており、特定の人ばかりが発言して、発言しない人は全く発言しない傾向が顕著に現れている。

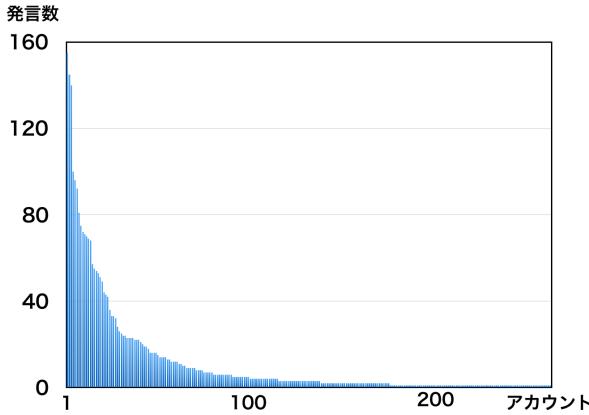


図 14 WISS2015 のチャットにおけるアカウントと発言数の分布

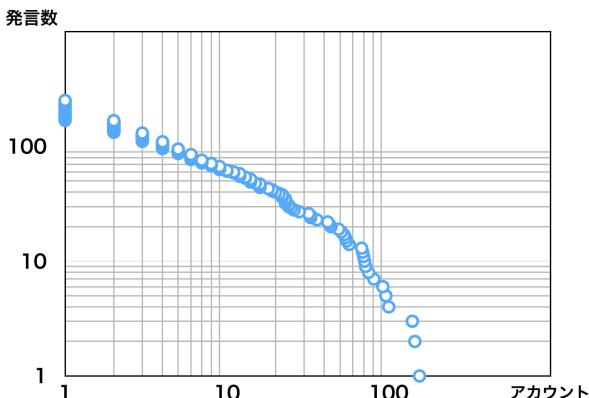


図 15 図 14 の両対数グラフ

『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、多数の人間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 16 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、教員、企業の研究者など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ってもらうことができると考える。



図 16 150 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実績

我々の研究室では、約 6 ヶ月の間『わかるらんど』を研究室内の大型ディスプレイに表示して実際に利用してきた。我々の研究室では WebLinda を使用して、Web 情報や Arduino や Raspberry Pi に接続したセンサ情報を利用し、

- 部屋の明るさを知る
- 部屋の温度を知る
- 風速と風向を知る
- 入口のドアの鍵を開ける
- 部屋の照明を点ける/消す

といったことを Slack^{*9} のチャットボットを通じて行える IoT 環境を構築している(図 17)が、『わかるらんど』と組み合わせることでさらに便利に使うことができた。



図 17 Slack のチャットボットによる IoT 機器の操作

図 18 は部屋の明るさを表示するデータセルだが、明るさのデータの値が小さい時は background 画像を変更することで、照明が点いているのか消えているのかがわかるようになった。

図 19 は、研究室のドアが最後に開いた時間を表示するデータセルである。普段は扉の画像を background にして

*9 <https://slack.com>

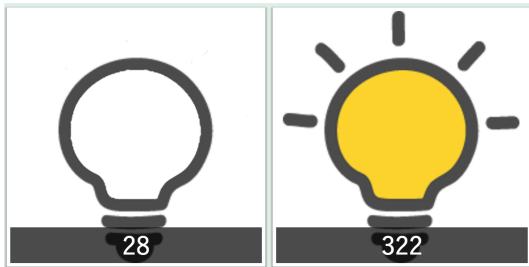


図 18 照明が消えている時（左）と点いている時（右）

いるが、実際に扉が開いた時に 10 秒間だけ background を図 19 右のように扉を開ける人の画像にすることで、誰かが入室したことが視覚的にわかるようになった。



図 19 ドアの普段の表示（左）と誰かが入室した時（右）

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、普段発言の少ない人でも何らかの反応を表明したり、発表者が聴衆の反応をひと目で把握することができるようになった。発表中に『わかるらんど』に「わからん」というリアクションが並んだときは、発表者が詳しい説明をすることができたり、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び、『わかるらんど』を通じて一体感が生まれたりすることもあった（図 20）。



図 20 『わかるらんど』を使用したミーティング

また、コンピュータで何か別の作業をしているときに、Web ブラウザで『わかるらんど』を開いてリアクションするのが面倒であるという意見もあった。そこで、図 21 や図 22 のような独自の『わかるらんど』入力ハードウェアを作成した。これらを利用することで、限られたリアクションではあるが、Web ブラウザを開くことなく『わかるらんど』にリアクションを表示することができるようになった。

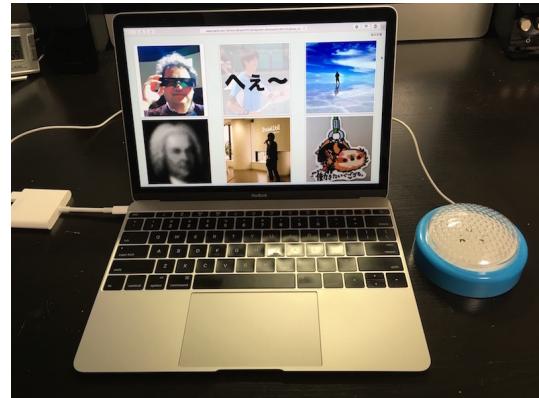


図 21 押すと「へえ～」とリアクションできるボタン



図 22 テンキーを利用した入力装置

5. 関連研究

情報ダッシュボードのデザイン [2] に関する研究は多くないが、表示すべき情報を選択する手法 [5] や、セルの自動配置手法 [4] などの研究が存在する。『わかるらんど』のダッシュボードは単純な形状のセルを指定どおりに並べているだけであるが、よりわかりやすい表示のための配置手法の検討は意義があると思われる。

会議での議論を促進するために「On Air Forum」[7]、「Lock-on-Chat」[8] など様々なチャットシステムが提案されているが、このようなチャットシステムのほとんどはタイムライン型式で表示が行なわれるようになっており、情報ダッシュボードのような型式で感情や意見など書き込んで一覧できるチャットシステムは存在しない。

近年、消極的な人間でも会議の議論などに参加しやすくなるための研究が消極性研究会 (SIGSHY: Special Interest Group on Shyness and Hesitation around You) というグループなどを中心に盛んになってきているが [6][7]、4.1 で述べたように、『わかるらんど』もこのような方向性の支援に利用できる。

6. まとめと今後の課題

本論文では、共有情報視覚化システム『わかるらんど』を提案した。『わかるらんど』は、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一覧表示し，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することができる。会議などで特定の人が沢山発言する状況を，本システムを使うことで，誰もが気軽に意見を表明できる環境を構築できる可能性を示した。また，研究室での6ヶ月の利用実績から，ネット上のあらゆる情報を投稿/共有するIoT時代の新しい情報視覚化手法として本システムが有益であることを示した。

今後は，授業やコンファレンスなどで大人数で『わかるらんど』を利用してもらう実証実験を行うことを考えている。

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。, ピー・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).
- [8] 西田健志, 五十嵐健夫：Lock-on-Chat：複数の話題に分散した会話を促進するチャットシステム, コンピュータソフトウェア, Vol. 23, No. 4, pp. 69–75 (2006).
- [9] 後藤真孝: WISS 2010 と WISS 2011 での改革, コンピュータソフトウェア, Vol. 29, No. 4, pp. 3–8 (2012).
- [10] 綾塚祐二, 河口信夫：参加者が作る会議支援システム：WISS Challenge(特集: インタラクティブソフトウェア), コンピュータソフトウェア, Vol. 23, No. 4, pp. 76–81 (2006).