

重さを物体識別 ID として用いるメモ共有システム

橋 本 翔[†] 増 井 俊 之^{††}

実世界と情報世界を接続し、大学の研究室内での情報共有を便利にするシステムを作成した。本システムには 3 つの機能がある。1 つ目の機能は、物体を識別する機能である。研究室内では各人が共有物を使って工作や勉強を行っている。また、私物を持ち込む場合もある。しかしそれらの物品は小型な場合が多く、RFID やバーコードなどのタグを付けると実際の作業の邪魔になってしまう。そこで本研究では、重さをを用いて物体を認識するしくみを作成した。2 つ目の機能は、物に対して web 上から複数人で情報を付与する機能である。研究室内にある機材には付箋や書き込み以上の注釈を書きたい場合もあるし、写真や映像で使い方を説明したい事もある。3 つ目の機能は、web 上での情報更新を実世界に通知する機能である。私たちは常にラップトップコンピュータを開いているわけではなく、移動したり、他の作業をしていたりする。一般的に研究室の全員に情報を共有する方法としてメーリングリストを使う事が多いと思うが、そうではなく web 上の情報を主として、その更新通知のみを行うもっと軽い方法が必要だと考えた。

Memo shareing sytem uses mass as ID to detect Object

SHO HASHIMOTO[†] and TOSHIYUKI MASUI^{††}

english english english

1. はじめに

実世界と情報世界を接続する時に、主に 3 つの問題がある。物と web 上の知識のリンク方法と、web 上でいかに知識を共同編集するかという事、そしてその更新をユーザーにどのようにして通知するかという事である。本論文で説明するシステムでは、大学の研究室という環境において工具や道具などの物体から web 上の知識へと簡単にリンクさせる事ができ、また web 上での知識を共同編集し、その編集過程を研究室のメンバーにリアルタイムに通知する事ができる。

大学の研究室にある特殊な工具についての情報を web 上の情報とリンクさせる際に、まずユーザにとって物の正確な名前がわからないので検索できない事が問題になる。その問題を解決する方法として（ここに関連研究：バーコードの例、RFID の例、画像検索の例、お茶大の WISS）本研究では、物の持つ物理的な特徴である重さをを用いて、物体を認識する事にした。

物体認識に重さを使う事には以下の 2 つの利点がある。1 つ目は、使いやすい事である。バーコードの様に正しい向きでスキャンする必要が無く、ただはかりに乗せるだけで良い。また画像検索と異なり正面を向けて置く必要もない。2 つ目は、物体が最初から持っている物理的特徴を利用するので、タグ等を貼り付ける必要が無い事である。

web 上での知識の共同編集方法の代表的な物としては wiki が挙げられる。MediaWiki¹⁾ 等の wiki には更新通知機能があるが、その通知方法はメールで定期的なまとめて通知されるものである。しかし、研究室のメンバーはそれぞれがパソコンに向かっていたり、移動していたり、電子工作をしていたりする。つまり既存の wiki とその更新通知システムでは、多様な状態を持つユーザに対してリアルタイムな更新通知ができない。本研究では、web サービスとして実装されたアウトラインエディタと、様々なメディアに適切なフォーマットで更新通知するシステムにより、この問題を解決した。

2. システムの概要

本システムには、1. 重さにより物体を認識する機能、2. 物について web 上で文書を共同編集する機能、3. web 上の文書の更新をユーザに適切に通知する機能

[†] 慶應義塾大学 政策メディア研究科

Keio University, Graduate School of Media and Governance

^{††} 慶應義塾大学 環境情報学部

Keio University, Faculty of Environment and Information Studies



図 1 重さを計るセンサーの試作版
Fig.1 prototype of sensor to detect weight

がある．ここでは，実装したシステムの使い方について説明する．

2.1 はかり

ユーザは研究室にある工具や道具をはかりに乗せる（図：1）はかりは重さを計測し，物体と重さのデータベースを検索する．該当する重さが存在すれば，はかりが接続されたコンピュータにウェブブラウザが表示される．データベース上に該当する物体が無ければ，新しく名前を付ける画面が表示される．

2.2 gyaazz

ウェブブラウザ上では gyaazz というアウトラインエディタが表示される（図：2）gyaazz は gyazz²⁾ を参考にして実装されたウェブアプリケーションである．マウスでクリックした行を画面遷移する事無くその場で編集でき，[と] の記号のみを用いたマークアップで簡潔に画像やリンクなどを挿入できる．emacs 風のキーボードショートカット機能を持ち，アウトラインエディタとしてのブロック単位の編集が容易になるようにデザインされている．複数のユーザが同時に 1 つのページを編集した場合も，編集内容は自動的に同期される．ユーザはこの gyaazz 上で物に関する情報を記述する．

2.3 gyazz-checker

gyazz-checker は gyaazz 上での編集をリアルタイムに他のユーザに更新通知できるツールである．gyazz および gyaazz の更新を，パソコン上のインスタントメッセージング，スマートフォン，twitter に通知できる．gyaazz はアウトラインエディタなので，更新の差分は行毎になる．gyazz-checker は数分おきにページの上から順に行毎に差分を取り，変更があった行と新規挿入された行をユーザに通知する．この行毎の差分が様々なメディアでの表示に容易に最適化できる事



図 2 ウェブアプリケーション gyaazz
Fig.2 gyaazz : outline editor web application

が重要である．

gyazz-checker はパソコン上で使うインスタントメッセージングに対しては，全ての変更を通知する．5 秒間隔で 1 行毎に送信する事で，まるでチャットのように受信される（図：3）また通知には im.kayac.com³⁾ も対応していて，iPhone や Android 等のスマートフォンにも push 通知を行うことができる．

gyazz-checker に twitter アカウントを設定しておくと，そのアカウントにも更新通知が行われる（図：4）しかし twitter に全ての差分の通知を行うとタイムラインが gyazz-checker で埋まってしまう．研究室メンバーはそれぞれ twitter 上で 300 から 2000 人を follow している．メンバーが煩わしさを感じずに gyaazz 上の更新の盛り上がりを twitter 上でも感じられる様に調整した結果，1 ページ毎に 1 行から 3 行を 10 分毎に twitter に通知する事にした．

3. システムの実装

本システムは 2 章で説明した通り，はかりとアウトラインエディタと更新通知システムの 3 つに分けられる．この章では 3 つを順に説明する．

重さを物体識別 ID として用いるメモ共有システム



図 3 google talk 上での gyazz-checker
Fig. 3 gyazz-checker on google talk

3.1 はかりの実装

ストレンゲージは体重計やキッチンスケールに使われる部品である．金属製の角棒の 2 面に抵抗が貼りつけられていて，棒のわずかなしなりに伴い貼りつけられた抵抗も伸縮する．2 つの抵抗値の差分を計測する事で，物の重さをはかる事ができる．

ストレンゲージはタニタ社製キッチンスケール KD-189 を分解して取得した．オペアンプ AD620BNZ で差動増幅回路を組み，その出力を Arduino⁴⁾ の 10bit AD コンバータで計測する．AD620BNZ のゲイン設定用抵抗には温度係数の小さい巻線抵抗を使用する事で，気温の変化による誤差を小さくする事ができた．計測値を USB ケーブルでパソコンに送信し，パソコン上の Ruby スクリプトで 100 回分の平均と値の標準偏差を取得する．パソコン上のプログラムは，初回起動時に 10 グラム，20 グラム，30 グラムの重さの分銅を乗せて値を学習させる事でキャリブレーションが行える．ストレンゲージ毎に個体差があるが，約 0.5 グラム単位で重さを計測する事ができた．100 回分の重さの標準偏差が 0.5 グラム以下の時，物体がはかりの上で静止していると認識し，重さと物体のデータベースに問い合わせる．データベースは key-value ストアである MongoDB⁵⁾ と，Ruby のウェブアプリケーションフレームワークである Sinatra⁶⁾ で実装されていて，HTTP と JSON による RESTful な API で扱うことができる．重さデータベースは重さに対して適切な



図 4 twitter 上での gyazz-checker
Fig. 4 gyazz-checker on twitter

gyaazz ページの URL を返すので，それをパソコン上のウェブブラウザで表示する．

3.2 gyaazz の実装

gyaazz は Sinatra で実装したウェブアプリケーションで，データベースには Tokyo Cabinet⁸⁾ を利用した．UI は jQuery⁷⁾ を利用して JavaScript で実装した．クリックした行毎にマウスイベントを割り当ててあり，HTML を form タグに書き換える事で行ごとの画面遷移なしの編集を実現している．キーボードの上下左右キーおよび emacs 風の ctrl キーと p,n,b,f,a,e キーを同時に押す事により，カーソルが移動する．編集集中の行で ctrl キーと上下左右キーを同時に押すことで，その行を上下の行と入れ替えたり，インデントする事ができる．shift キーと同時に押すとアウトラインエディタとしてブロック毎に入れ替えたり，インデントを変更できる．ユーザが 20 秒間操作していないか，ウェブブラウザのウィンドウが非アクティブになった時に，行の編集状態が解除される．この時サーバーと通信してデータを保存し，ウェブブラウザ上の表示もサーバーから取得した最新の物に更新する．これにより，複数人での同時編集が可能となっている．

3.3 gyazz-checker の実装

gyazz-checker は Ruby スクリプトで実装されたツールで，gyazz および gyaazz のページを全件取得し，更

新のあったページとその内容をスマートフォンやインスタントメッセージ、twitter に通知する。取得したページは Tokyo Cabinet に時刻をキーとして蓄積され、一回前の取得データと

wiki の更新は定期的に twitter と iPhone と Android と IM に流れる。便利!! github へのリンクを貼りまくろう。はかりとか更新通知の精度・速度については適当に書く。

4. ま と め

重さで物体認識して web 上の情報と関連付け、wiki も作って、通知機能も作った。便利である。

参 考 文 献

- 1) MediaWiki. <http://www.mediawiki.org>
- 2) Gyazz. <http://gyazz.com>
- 3) im.kayac.com. <http://im.kayac.com>
- 4) Arduino. <http://arduino.cc>
- 5) MongoDB. <http://www.mongodb.org>
- 6) Sinatra. <http://www.sinatrarb.com>
- 7) jQuery. <http://jquery.com>
- 8) TokyoCabinet. <http://fallabs.com/tokyocabinet/>