3E3-OS20-7

集合知データベースWedataの構築と運用

Constructing A Collective Intelligence Database Wedata

江渡 浩一郎[*1] 沢田 洋平[*1]
Koichiro ETO Youhei SAWADA

[*1]独立行政法人産業技術総合研究所 社会知能技術研究ラボ
Social Intelligence Technology Research Laboratory, National Institute of Advanced Industrial Science and Technology (AIST)

We have constructed a collective intelligence database, "Wedata," that enables us to create and edit data in order to run programs collectively. Wedata currently has 146 datasets and more than 53,000 items. With the increase in usage, we have logged more than 24M accesses from 1.2M unique IPs in December 2010. In this paper, we detail the knowledge we have attained from developing and operating Wedata.

1. まえがき

我々は，プログラムの動作に必要なデータを Web サイト上で共有し，共同でデータを追加・編集する集合知データベース *Wedata* を構築・運用している．これは，データベースの構築・管理に Wiki のような利用者参加型の設計手法を応用したものであり，この手法を *DataWiki* **方式**と名付けている．

Wedata では，対応するアプリケーション毎に 1 つのスキーマを定め，利用者はそのスキーマに従ってデータを追加・編集できる．アプリケーションは，必要なデータをデータセットとして一括して取得し，利用する．利用者は自分が必要なデータが存在していない場合には自分で追加でき，アプリケーション制作者はデータの追加・更新を利用者に委ねることができる．このように，アプリケーションが使うデータを利用者と共に成長させることによりアプリケーション制作者と利用者が共に利益を得る共生プロセスを実現することができる．

DataWiki 方式及び Wedata は，われわれが独自に考案したものだが，2006 年に Tim Berners-Lee が提唱した *Linked Data*[Berners-Lee 06] に関係が深い．Linked Data はデータの公開手法を定めた一種の設計指針であり，各々のデータに URI を付与し，通常の HTTP でデータを取得できるようにし，データ形式には XML などの標準的な形式を用い，関連するデータへの連携をリンクで表現するといった特徴を持つ．これは，Wedata の特徴と一致しており，Wedata は Linked Data の 1 アプリケーションと考えることができる．同時に，Wedata は利用者参加型でデータベースを構築する実践例であり，このような利用者参加型プロセスによるデータベース構築事例は Linked Data のような大規模データベースを構築する時に利用者参加型プロセスを用いる際の参考になると考えられる．

本論文では，Wedata の構築・運用における知見及び，Linked Data との関連について述べる．

2. DataWiki 方式

2.1 DataWiki 方式の構成要素

様々な Web サイトの情報を扱うアプリケーションを設計する際に，サイト毎に異なる処理が必要になることがある．最も単純に実装する方法は，if 文などの条件文でサイト毎に異なる記述をすることである．しかし，対応サイトが増えたりサイトの構造が変化すると，元のプログラムを書き換える必要がある．

このような場合，一般にプログラムとサイト毎の対応情報を分離し，対応情報をデータとして外部化して，プログラムは実行時にそのデータを読み込み実行するようにする．こうすると，元プログラムを変更することなく，データを追加・修正することで，更新に対応できるようになる．

しかし，誰かが対応情報を更新し続けなくてはいけない状況には変わりない．対応するサイト数は増えれば増えるほど有益だし，Web サイトの構造はしばしば変化するため，対応情報の更新には手間がかかる．対応情報の修正を行えるのがプログラムの開発者だけだとしたら，プログラムの引続き開発者に負荷が集中してしまう．

この時，このサイト毎の対応情報を，Wiki の手法を参考にし，インターネット上で共有して，誰もが編集可能な状態に置く手法を考案し，DataWiki 方式と名付けた．DataWiki は，以下の要素から構成される．

データベース Web 上でデータを共有・公開し，編集機能を提供するサーバプログラム．

データセット 特定用途のために用意されたデータの集合．一般に，1 アプリケーションにつき 1 つのデータセットが定義される．

アイテム アプリケーションが使うデータの最小単位．キーとバリューの組み合わせによって構造化される．キーとバリューは任意長の文字列とする．

スキーマ 1 つのデータセットに属するアイテムがどのようなキーを持ちうるかを決める．1 つのデータセットにつき 1 つのスキーマを持つ．データセットの制作者のみが変更できる．

クライアント データベースにアクセスしてデータセットを取得し，利用するアプリケーション．

2.2 DataWiki 方式の基本思想

各々の要素の連携を元に，基本思想について解説する．

連絡先: 江渡浩一郎，独立行政法人産業技術総合研究所社会知能技術研究ラボ，〒 135-0064 東京都江東区青海 2-3-26，k-eto[at]aist.go.jp

データ作成・編集への参加を促す DataWiki 方式は，アプリケーションの利用者がデータベースに各々が必要なデータを入力することによって成り立っている．そのため，ユーザが気軽にデータを入力できるよう，連携するシステム全体に気を配る必要がある．データの追加・編集を誰もが行えるようにしたとしても，データ作成の難易度が高すぎると，結局データ作成を行うユーザは限られてきてしまう．できるだけ，必要な知識や技術を少なくし，敷居が低くなるように設計する必要がある．

データの安全性を確保する データベースに蓄積されるデータは，不特定多数のインターネット上のユーザが入力したものであるため，データの信頼性が問題になる場合がある．例えば，スクリプト片のようなプログラムの一部を共有してしまうと，アプリケーション側で安全性の検証が必要となる．追加・編集されたデータを少数の権限を持つ人が確認した上で反映されるようにする方法もあるが，確認作業に負荷が集中してしまい，遅延が生じ，その遅延を利用者が嫌って追加・編集しないという悪循環が発生してしまう．そのような確認作業が必要無いように，外部から静的に安全性を確認できる安全なデータだけを共有するようにする．

最低限のアクセス管理機構 誰でも追加・編集・削除ができるとしても，スパムや望まれない編集機能の悪用は禁止したい．スパムなどの編集機能の悪用を放置すると，入力されたデータを破壊されたり，意味のないデータが大量に投稿されるなどの問題が生じ，有意義な共同編集が妨げられる．それらを防止できるよう最低限のアクセス管理機能を用意したいが，システム独自にアカウントを登録させると，利用者にアカウント管理の負荷が発生する．そのため，OpenID などにより他の既存のシステムを用いたユーザ確認を行う．

編集活動の可視化 追加・編集活動は，単なるデータの更新ではなく，他の利用者への伝達も含めた社会的な活動である．あるデータセットに誰がデータを追加・編集したのかが他の必要な利用者に伝わるように可視化する．追加・編集活動を RSS などのフィードで参照できるようにして，利用者間で随時編集活動の相互確認を行えるようにする．

API の整備 Web ブラウザからのデータの追加・編集だけでなく，プログラムからのインタフェース (API) を整備することにより，プログラムからも追加・編集などを行えるようにする．

データの権利の扱い 登録されるそれぞれのデータは，利用者が独自に考案して作成したデータであり，そのためデータの権利はその制作者に属する．しかし，集合知データベースに集められたデータは一般にはデータセットとして集約されて初めて意味があるデータである．そのため，データ制作者の意志と集合知データベースとしての価値を同時に満すために，データの権利の扱いについて事前に明示しておく必要がある．データを登録する前のアカウント登録時に登録されたデータの権利を明解化して同意を得る．

利用者の編集活動への参加促進 クライアントは，不特定多数のユーザによるデータセットによって成り立っていることを明示する．アプリケーションの配布サイトやヘルプ

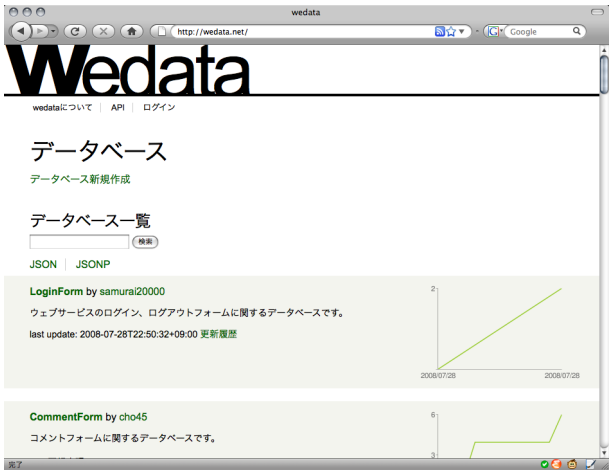

図 1: Wedata のトップページ画面

ドキュメントなどで，共有のデータセットを使用していることと，それがどのようなものであるかを明記しておく．データの追加・修正を行うと，そのアプリケーションにどのような効果があるかを利用者に伝える必要がある．

2.3 SITEINFO の設計

DataWiki が保持するデータセットのうち，特に Web サイトの構造情報を保持するデータセットを *SITEINFO* と呼んでいる．SITEINFO は，データセットのスキーマの一形態であり，Web サイトの構造情報を指定するためのいくつかの要素を持つ．SITEINFO の個々の要素が持つ属性は，クライアントの要求によって異なる．最低限必要な要素は，適用可能な URL を指示する正規表現やワイルドカードである．Web ページの構造情報属性には，XPath，CSS セレクタ，正規表現などが用いられる．その他，適用可能な URL の実例を保持しておくと，どの URL でそのアイテムを適用できるかがわかるため，自動的な検査機構を実現できる．

3. 実装

DataWiki 方式の構成要素であるデータベースと，クライアントの実例について述べる．

3.1 Wedata の実装

DataWiki 方式に基いた集合知データベースとして，Wedata[*1](図 1) を構築した．これは DataWiki 方式に基く集合知でのデータ収集を実現するべく，立ち上げたインターネット上の Web サービスである．

アプリケーション制作者は自分が使うデータセットを自由に作ることができ，そのデータセットのスキーマを定義できる．他の利用者は，ログイン後に誰でも要素を追加・編集・削除できる．データセットの作成や要素の追加・編集・削除には，Wedata へのアカウント登録が必要であり，登録・認証には OpenID を使っている．アカウント登録時に，登録されるデータの権利の扱いへ同意することとなっている．

また，Web 上のインタフェースのみならず，プログラムからデータを読み書きするための API を提供している．データの読み込みは誰でも可能であり，データセットを JSON 形式 json で一括で取得できる．データセットの作成や要素の編集

∗1 http://wedata.net/

表 1: AutoPagerize の SITEINFO の属性

属性	必須	内容
url	○	適用する URL を示す正規表現
nextLink	○	次のページの URL を持つ要素を指定する XPath 式
pageElement	○	継ぎ足しするページ本体を指定する XPath 式
exampleUrl		適用される URL の一例
insertBefore		次のページを挿入する箇所を示す XPath 式

は，Wedata のアカウント登録が必要であり，登録語に発行されるトークンを用いて認証する．

3.2 クライアントの実例

クライアントの実例として，筆者の一人が開発した *AutoPagerize*[*2]の実装について述べる．AutoPagerize は，様々な Web サイトで**ページの自動継ぎ足し**[*3]を実現する Greasemonkey スクリプトである．現在では Firefox だけではなく，Google Chrome や Safari などでも使えるようになっている．

ページの自動継ぎ足しとは，Web ページの閲覧中に次ページを自動的に読み込み，その内容をページの下部に継ぎ足して表示する機能のことである．例えば Google の検索結果のように，Web ページ中に「次へ」のリンクで次ページが示される場合がある．この時ページ下部までスクロールすると，裏側で自動的に次ページが読み込まれ，現在読んでいるページの最下部に挿入される．結果として，それぞれのページが縦につながった 1 つの巨大なページのように見える．

AutoPagerize で使用しているデータセットのスキーマを表 1 に示す．適用可能な URL を柔軟に表現できるように url は正規表現で持っている．Greasemonkey スクリプトでは，Firefox に実装されている XPath 処理系を利用できるため，nextLink，pageElement，insertBefore では，ページの特定の個所を柔軟に指定できる XPath を値として持つようにした．実際に使用されている SITEINFO の例として，Google 検索の SITEINFO を表 2 に示す．

Google 検索結果の例では，url は www のようなサブドメインの有無や，国ごとに違ったドメインが使われている場合でもマッチする正規表現となっている．nextLink と pageElement は，それぞれページ上の該当する要素を示す XPath 式となっている．exampleUrl には，AutoPagerize という単語を検索した場合のページが示されている．自動継ぎ足しの動作を実現するには，nextLink と pageElement の要素を正確に取得できることが必要不可欠である．そのため AutoPagerize では，SITEINFO 方式により人手で作成された XPath 式をもとに取得している．

AutoPagerize は，1 日 1 回 Wedata から最新のデータセットを取得するようになっている．Wedata では，日々，利用者によるデータの追加・修正が行なわれており，それにより様々な Web サイトへの対応が実現されている．

4. 運用実績

4.1 Wedata 運用実績

Wedata は 2008 年 3 月にアルファ版を公開開始し，2008 年 4 月に一般公開，2011 年 5 月現在も運用中である．現在，146 件のデータセットが登録されている．アイテム総数は合計 53,000 件以上となる．

*2 http://autopagerize.net/

*3 自動継ぎ足し機能は，GoogleAutoPager[mala 05] を元にしている．

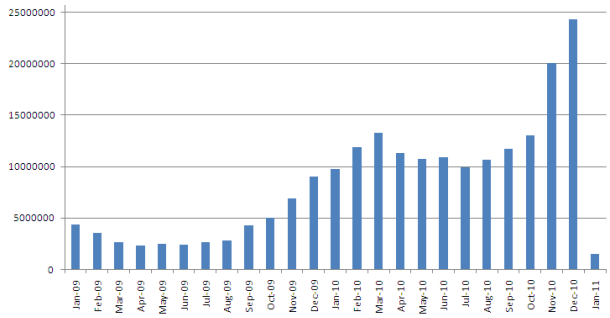

図 2: Wedata の月間アクセス数のグラフ

図 2 に月間アクセス数のグラフを示す．2010 年 12 月のピーク時には 2,427 万件のアクセスがあった．このグラフに示すように，2010 年 8 月以降に大きなアクセス数の増大が見られた．これは，Google Chrome 用の AutoPagerize 実装である AutoPatchWork[*4] が Google Chrome のトップページからおすすめとしてリンクして公開されたことにより，多数の利用者が AutoPatchWork 経由で Wedata を使うようになったためと考えられる．

5. 関連研究

5.1 経験則との比較

Web ページの構造が比較的簡単な場合には経験則（ヒューリスティクス）で扱える．例えば，リンク内の文字列に「次へ」という文字列が含まれている場合は，次ページへのリンクである可能性が高いと判定できる．*Opera*[*5]では，この手法によって次ページへのリンクを検出している．この手法は人工知能的なアプローチと同じであり，事前にサイト毎の対応情報を用意しておかなくても，多くの場合は構造を検出できるという利点を持つ．しかし，常に検出できるわけではない．自分が使いたい Web サイトが対応していなければ，プログラムが改良され，検出アルゴリズムが改善されるのを待つしかない．これに対して，DataWiki 方式では，対応情報を利用者が明示的に指定するため，常に検出できる．対応していないサイトがあっても，利用者が自分で情報を追加できる．サイトの構造が変化したり，データに誤りがあったときは，利用者が自分で修正できる．

また，経験則と DataWiki 方式とは相反する方式ではなく共存できる．一般には経験則で自動的に処理し，高度な対応が必要な場合のみ DataWiki 方式を使うとして並用できる．

5.2 セマンティックウェブ，microformats，Linked Data

Web 上のデータを再利用可能なものにしようという試みは，長年に渡って続けられている．Tim Berners-Lee は，**セマンティックウェブ**[Berners-Lee 01] において，Web 上のデータに意味的なメタデータを付与して，機械が情報を理解できる Web の世界を目指した．メタデータ付与のための *RDF*[Lassila 99] という規格が策定され，継続的な努力によって利用が拡大しているが，いまだに一般的な存在になったとは言いがたい．

RDF の利用には高度な技術的知識が必要であり，手軽には扱えない．そのため，より簡易にメタデータを指示する *mi-*

*4 `https://chrome.google.com/webstore/detail/aeolcjbaammbkgaiagooljfdepnjmkfd`

*5 http://www.opera.com/

表 2: Google 検索用の AutoPagerize の SITEINFO

属性	値
url	`^http://[^.]+\.google\.(?:[^.]+\.)?[^./]+/(?:search|custom|cse)`
nextLink	`id("navbar")//td[last()]/a | id("nn")/parent::a`
pageElement	`id("res")/div[div]`
exampleUrl	`http://www.google.com/search?q=AutoPagerize`

croformats[*6]という記法が提唱された．HTML 中に人間が読める形式でメタデータを埋め込むことで，比較的容易にメタデータを付与できるようになったが，サイトの運営者が対応する必要がある点は変わらない．

2006 年に，Tim Berners-Lee が提唱した Linked Data は，これまでセマンティックウェブで培ってきたメタデータの取り扱いに関する仕組みを発展させて，さまざまなデータをウェブ上で公開して共有しようとする仕組みである．全てのデータを URI で参照可能にし，その URI に HTTP でアクセスすることによってデータを取得できるようにし，データ形式を XML などの標準的な記法として，関連するデータへのリンクを同じく URI で含めるようにするという，ウェブ上でのデータ公開における設計指針である．

我々は Wedata を 2008 年に公開開始しており，当時は Linked Data について知らなかったため，独自に DataWiki 及び Wedata の仕組みを考案しているが，結果的にはこれらの Linked Data の特徴は Wedata の特徴とよく似ている．さまざまなデータをウェブ上で公開・共有する仕組みであり，全てのデータは一意の URI を持ち，HTTP でアクセスすることによって，JSON 形式で取得できる．

しかし，違いもある．DataWiki 方式は，何よりも Wiki という特徴を継承している点にある．Wiki は利用者参加型でコンテンツを作るという特徴を持つ．これは，単に誰でも編集できれば良いというわけではなく，多数の利用者が 1 つの成果物を共同で育てていくためには，利用者参加型設計プロセスにおいて，関係者が運用ポリシーやルールを共有しておく必要がある [江渡 09]．これによって初めて共同での成果物の制作が可能となる．

集合知データベース Wedata は，このような Wiki における利用者参加型のプロセスと Linked Data のようなデータ公開手法を組み合わせたものと理解できる．そのため，現在 Linked Data において行われている取組みにおいて，集合知の仕組み，つまり利用者参加型のプロセスを応用することができるだろうと考えている．

5.3 集合知によるデータベース構築の取り組み

2010 年 12 月に，Google は Google DataWiki[*7]というサービスを開始した．DataWiki は，構造化されたデータ (複数の項目を持った表) を扱うための Wiki システムである．通常の Wiki を拡張して，構造化データを簡単に作成・編集・共有・可視化できる．ラベルと文字列によって保持される構造化されたデータを，Wiki のように誰もが追加・編集・削除できるデータベースと説明されており，これは DataWiki 方式の要素とほぼ重なっている．

しかし，目的が大きく異なっている．DataWiki は災害時に救助に必要な情報を素早く集積させる目的で設計されている．そのため，緯度経度情報のような地理情報を集約でき，地図上の地点として可視化できる．それに対して Wedata はウェブサイトの構造情報などといったさまざまなアプリケーションから使えるデータをインフラとして長期に渡って提供することが目的である．その他の違いとして，Wedate は JSON や JSONP でデータ取得可能，OpenID を使った認証，外部 API からデータセットの作成，アイテムの変更，削除，検索などといった操作が可能といった違いがある．

しかし，元々の DataWiki というコンセプトがまったく同じであり，同種のサイトが登場してきたことに意を強くしている．

6. まとめ

インターネット上の不特定多数の利用者の参加によってデータベースを構築する手法である DataWiki 方式を提案した．DataWiki 方式に基いた集合知データベースの Wedata を構築し，運用している．Wedata では，AutoPagerize を含む 146 件のデータセットが登録され，現在もデータが増え続けている．今後，さらに DataWiki 方式の活用で，様々なデータベースが集合知によって構築されることを期待する．また，Linked Data の取組みにおいて，DataWiki 方式の知見を生かせるように発展させていきたい．

謝辞

本研究は，ngi group 株式会社との共同研究の一環として実施されました．感謝の意を表します．また，Wedata は利用者と共に作り上げるデータベースです．データベース構築に協力していただいている皆様，またデータを利用するアプリケーションを開発していただている皆様に，心から感謝の意を表します．

参考文献

[Berners-Lee 01] Berners-Lee, T., Hendler, J., and Lassila, O.: The Semantic Web, *Scientific American*, Vol. 284, No. 5, pp. 28–37 (2001)

[Berners-Lee 06] Berners-Lee, T.: Linked data, *International Journal on Semantic Web and Information Systems*, Vol. 4, No. 2 (2006)

[江渡 09] 江渡 浩一郎：パターン，Wiki，XP：時を超えた創造の原則, 技術評論社 (2009)

[Lassila 99] Lassila, O., Swick, R., et al.: Resource Description Framework (RDF) Model and Syntax Specification, *W3C Recommendation* (1999)

[mala 05] mala, : GoogleAutoPager (2005), http://la.ma.la/blog/diary_200506231749.htm

∗6 http://microformats.org/

∗7 http://datawiki.googlelabs.com/