

第 2 章 「Wiki」って何？

大部分の人には、「Wiki サーバー」という用語は理解できないものです。その意味するところを説明しようとしても、コンセプトがとても単純ながら奇抜でもあるので、把握しにくいのです。このことは、いくつかの点で自転車に乗ることに似ています。つまり、どのようなものか分かっている人にとっては簡単かつ自然なもので便利な手段ですが、一度も経験がない人にとっては、見たところ不合理に思えるものなのです。

それは承知した上で、この章では、前章で触れた共同作業や議論を行うサーバーの概要から一步踏み込み、主要な機能の特徴を紹介し、バリエーションの実装に言及しながら Wiki のコンセプトを解説します。この章をあまり理論寄りにしないために、ユーザーの体験例やユーザーコミュニティから得られた声を参考にして、いくつかの重要なポイントを説明していきます。

2.1 この章の内容

この章では、Wiki 技術のコンセプトについて解説し、ウェブ上の共同作業に Wiki や Wiki ライクなものを利用するソリューションについて、より詳細な解説を行います。

最初の主要なセクションである『Wiki のコンセプト』では、Wiki 自体に焦点をあて、その歴史と基本的な特徴を見ていきます。『Wiki のエッセンス』では、Wiki をとてもユニークなものにしている特徴について、より掘り下げた全体的な洞察を試みています。いくつかの実例を用いて説明するために、『ユーザー体験』では、初心者の視点から見た場合の、標準的な Wiki について解説を行います。『有用性の基準』では、Wiki をユーザーに役立つものにする要素を明らかにし、さらに事例の解説を行います。『Wiki の原理』においては、Wiki の基本部分をオーサリングするのに欠かせない「仕組み」をまとめます。

次の主要なセクションである『Wikiクローン』では、Wikiの基本コンセプトが、異なる環境、言語、要件にどのように適用されてきたかを示すために、数ある実装バリエーションの中からいくつか紹介し、対象範囲を広げます。『プログラミング言語によるWikiの実装』では、表を使って、Perl、Squeak、そしてRubyによる実装についてさらに検討を行い、種々のクローンの概要を見ていきます。『その他のWiki環境の提供』では、既存のWikiホスティングサービスの例を示します。『Wiki以外のサーバー』では、Wikiの特徴を1つ以上共有する、類似した双方向サーバーのコンセプトについて解説します。

最後に、『Wikiアプリケーション』のセクションでは、Wikiを色々な状況下で動かす際に生じる問題を取り上げます。『Wiki型サーバーの長所と短所』では、一般ユーザーの操作性の観点から、実際にWikiを利用した場合の強みと弱みについて、バランスのとれた解説を試みます。『Wikiの立ち上げを検討する理由』では、Wikiを導入する根拠をいくつか示し、成功に結び付く要素を指摘します。『その他の問題』では、たとえばプレーンテキストの問題や、Wikiが「どこでも動く」リソースという理想をいかにして実現するかについて扱います。

2.2 Wikiのコンセプト

ハワイ語の用語集には、このような単語の定義が載っています。

Wikiwiki(状態動詞) 速い、迅速な。急ぐ、急がず。素早い。

この言葉は、ハワイでは、フォーマルな文脈でもカジュアルな文脈でも、単に「素早い」とか「形式ばらない」という意味でよく使われます。

WikiWikiWebサーバーコンセプトは、Ward Cunninghamが発案したもので、大抵の場合単に「Wiki」と呼ばれます。Wikiは、自由に拡張可能な、連結されたウェブ「ページ」の集積で、情報を格納し、修正するためのハイパーリンクシステムです。入力形態としてウェブブラウザ・クライアントを使うことで、各ページが誰にでも容易に編集可能になるデータベースなのです。

Tip 2.1 「Wiki」の表記

(編集部注)原書では、本質的な概念を「Wiki」、特定の実装を「wiki」と表記していますが(Perlも同様)、既存のPerl関連書等にならない、コード、URLなどを除くすべてを「Wiki」(Perl)で統一しました。

今日至るところにある同種のアプリケーションを利用し、Wiki コンテンツのナビゲート、閲覧、そして編集を行います。追加**アプレット**(大きなアプリケーションと組み合わせて機能するようコーディングされた小さなアプリケーション)や余分なものは何も必要ありません。つまり、ベース Wiki の機能は、現在のウェブブラウザで利用できる一般的な機能に限定されます。

Ward は Wiki を、「**できる限りの効果を発揮しうる、最も単純なオンラインデータベース**」と呼びました。1994 年、彼はウェブ上でソフトウェアパターンを共同で公開する、手取り早い方法を必要としていました。プログラム開発に関する彼の仕事から発展したアイデアと HyperCard^[1] スタックが組み合わさり、最初の「Wiki サーバー」が生まれました。Wiki は編集を簡単かつ手早く行えるので、ユーザーを惹き付けています。その上、Wiki の名称生成規則に従えば、有意義な形でページが自動的かつ的確に連結されます。

投稿により形成された**機構**を編集できるという点において、Wiki はグループコミュニケーションの手法としては珍しい存在です。Wiki と比較すると、電子メールやニュースグループへの投稿は、送信時に規定される種々の属性(作者、日付、題名)により自動的に組織化されます。閲覧クライアントの中には、メッセージ投稿に返信がついたものに注目し、題名により投稿をスレッド化して、さらに組織化を行うものもあります。読者は投稿の組織化を行う属性を選択できますが、追加情報をやり取りして、その構成をさらに改良することはできません。構造は固定化されているのです。Wiki は、コンテンツによる任意に変更可能な「有向ネットワーク」(ハイパーテキスト)機構をサポートします。

オリジナルの WikiWikiWeb のサイトである、Portland Pattern Repository(PPR、<http://c2.com/cgi/wiki>にあります)は、本書を執筆している時点で、およそ 13,000 もの数に上る「ページ群」により構成されています(月ごとに 500 ページ以上増えているようです)。そのページでは、刻々と変わる相互リンクが興味深い関係を定義しています。ソフトウェア設計手法について数多くの面白い議論が集まっており、また Wiki に関する議論と情報源のリンクも提供します。この何年にもわたって持続している Wiki は、多くの来訪者に影響を与え、創作意欲をかきたててきました。図 2-1 は、多くの場合「FrontPage」と呼ばれるデフォルトのトップページをキャプチャーしたものです。

多くの単純なコンセプト同様、「開放的な編集」というコンセプトは、Wiki の利用法に重要で微妙な影響を与えています。ウェブの民主的な利用を促進し、技術者でないユーザーによるコンテンツ作成を奨励することで、普通のユーザーがウェブサイトにどんなページを作成することも編集することも許可するのは刺激的なことです。

Wiki は変わっています。それは間違いありません。ある意味、ショッキングなくらい変わっています。ここでは、ウェブ上の Wiki ページで見かける様々な意見を要約し、その違いの一部について概説を試みます。



図2-1 オリジナルにして最大規模のWikiであるPortland Pattern Repositoryにおける、デフォルトの「フロントページ」

2.2.1 Wikiのエッセンス

技術的な見地からいうと、Wikiは世界的なインフラとしてのワールドワイドウェブと、偏在するサーバー・クライアントアプリケーションに支えられています。その基盤となるHTTPプロトコルは、クライアント・サーバー間の通信がどのように行われるかを規定します。Wikiは、このプロトコルのGET(データ要求)処理とPOST(データ提出要求)処理を行います。

ユーザーが目当たりにする機能レベルにおけるWikiのエッセンスは、以下の文章に要約できます。

Wikiはすべてのユーザーに、Wikiウェブサイト内のページを編集したり、新規ページを作成することを促します。そうした作業を行うのに、追加アドオンをまったく必要とせず、ごく普通のウェブブラウザのみを利用します。

Wikiは、ページ間のリンク生成をほとんど直観的に行えるくらいに簡単なものにし、ターゲットとするページが存在するか否かを表示することで、異なるページ間に有意義な話題の関連付けを行うことを推奨しています。

Wikiは、通りすがりの訪問者を特に考慮して作られてはいません。その代わりに、Wiki

は訪問者を、絶えずウェブサイトの見た目を変化させる、進行中の創作プロセス、共同作業プロセスに巻き込もうとします。

Wikiは、完全な自由、アクセスや利用のしやすさ、シンプルで統一性のあるナビゲーション規則を持ち、なおかつ形式ばった構造を必要としません。変わってはいるものの共同作業用空間の一種なのです。Wikiはまた、知識の組織化、相互連結を行う手段でもあり、おそらくそれが1人でWikiを利用する場合の主な用途になります。

Wikiは本質的に民主的です。どのユーザーも、他のユーザーとまったく同じ権限を持ちます。Wikiにより、アカウントとパスワードを入力する手間をかけることなく、ウェブ上での共同作業が可能になります。表面的には、修正可能なデータを管理する方法としては極めて危険な手法に見えるかもしれませんが、セキュリティ機構がない場合でも、実際のところWikiコンテンツはほとんど損害を被らないことが経験上明らかになっています。

ユーザーは、特定のWikiの基盤を成すメカニズムやストレージモデルに関する知識をまったく必要としません。つまり、ユーザーは自分のブラウザでやるべきことをやるだけでよいのです。

Tip 2.2 「ベース」Wikiって何？

本章のような一般的な議論において、我々は自分たちが指すものを、短く「ベース」Wikiと表記しています。いいかえるなら、それはオリジナルのWikiWikiWebのように完全にオープンで、なおかつ本書の後の章で議論するWikiのように最小限の機能セットを備える基本実装のことです。セキュリティ機能を付加する形でWikiをカスタマイズすると、そのWikiは容易に独裁的で非民主的なものになる可能性があるのです。

次のセクションでは、ここで述べたことが現実の問題として何を意味するか、ユーザーの体験例を用いて解説します。

2.2.2 ユーザー体験

典型的なWikiWikiWebとの最初の出会いについて、例を挙げて説明してみましょう。ウェブサーフィンに慣れていて、Wikiのコンセプトを目新しく感じた本書の読者は、ウェブブラウザを立ち上げ、本文中に示されているウェブアドレス(URL) たえば、`http://c2.com/cgi/wiki` のいくつかを指定してみたことがきっとあるはずです。そうすると(インターネットへの標準的な接続性は確保されているとします)、以前に示した図2-1に似たコンテンツがウィンドウ内に表示されます。

Start where we [WelcomeVisitors](#). It tells what we are doing, how to get around, and how you get involved. Frequent visitors may want to go straight to [RecentChanges](#), [QuickChanges](#), or another of the useful [StartingPoints](#).

Ready to add to the WikiWikiWeb?

The [TextFormattingRules](#) are very simple, and the [TipsForBeginners](#) will help you learn to apply them gracefully. You'll probably want to start by enlarging or editing Wiki pages that already exist. The [WikiWikiSandbox](#) is set aside for practicing with the editor – go there now and try it out. Once you've made a few trial updates there, you'll be ready to participate in discussions on the Wiki, and before you know it, you'll be [AddingNewPages](#).

What Categories of pages exist?

Read [WikiCategories](#).

What is on Topic?

Read [OnTopic](#), [OffTopic](#), etc.

[EditText](#) of this page (last edited [March 24, 2002](#))

[FindPage](#) by searching for browse [LikePages](#) or take a [VisualTour](#)

図 2-2

FrontPage における、PPR を初めて訪れた人のためのハイパーリンク

今ではよく知られているナビゲーション慣習として、テキスト内のいわゆるハイパーリンクは、下線を引いた形式で表示されたり、マウスポインタが有効なリンクの上を通ると、その色・スタイル・ポインタのバリエーションが変わったりします。

Wiki ページは概してそうしたハイパーリンクをたくさん含むので、図 2-2 のスクリーンショットに見られる、例ページの下部に集まっているリンクについて特に言及しておきます。

これらの有用なリンクが、クリックすれば Wiki コンテンツの閲覧、編集作業を行うページに到達する経路を提供しています。文章の文脈に基づいてキャピタライズされた(先頭を大文字にした)単語をつなぎ合わせると Wiki ページのリンクが作られる手法のおかげで、通常各リンクが指し示す内容は明白です。

特に、どのページにもあって目立つ EditText(テキスト編集)のリンク以外にも、コンテンツを追加する場合にはトップページの案内文に気を付けてください。訪問者は、最初 WikiWikiSandbox を訪れるように促され、実際のページコンテンツの修正を試みる前に練習を行います。TextFormattingRules や TipsForBeginners といった役に立つページへのリンクも、FrontPage に加えられています。

Tip 2.3 「補助」ウィンドウの利用

後で再度参照するのに便利な、一種の「補助」テキストを見られるように、関連リンクについて1つ以上の新しいブラウザウィンドウを開いておくことをお勧めします。副次的な話題や相互参照を辿る場合も、複数のウィンドウを開くようにすれば、Wiki の閲覧や編集の経験を早く積むことができます。



図 2-3

WikiWikiSandboxの編集

「サンドボックス」について、図 2-3 に 2 つのウィンドウを示しています。1 つは閲覧を行うページ、そしてもう 1 つ、編集フォームに同じページを展開したものを重ねて表示しています。このサンプルが、「何か編集を試そうとする際に、編集内容を確認しながら作業する」典型的な例です。

単純なテキストパターンによる Wiki の「マークアップ」のスタイルや、別ページへのハイパーリンクを分かりやすく定義する WikiWiki の手法に注目してください。訪問者がいくつか変更を加え、Save ボタンをクリックすると、場合によっては「Thank You for Edit (更新してくれてありがとう)」ページをクリックしてやりすごした後に、当該ウィンドウに更新されたページが表示されます。同じページを閲覧している他の誰もが、変更が投稿されるとすぐに、その更新内容を確認できます。

我々がモデルとして考える訪問者は、Wiki がどのようなレンダリングを行うのかを知るためにテキストを入力してしばらく時間を費やした後で、詳しい情報を提供しなければと感じた面白味のあるページを閲覧します。サンドボックスにおいて編集方法を教わっていれば、編集フォームを引っ張り、望ましいコメント文を挿入してからセーブするのは簡単なことです。

Tips も閲覧してもらえば、今寄稿者になった訪問者 (たとえば Jane という女性とします) は、「JaneVisitor」という名前から形成されるページリンクの形式で、「署名」を加えることを理解してくれます。それを行うと「?」リンクが表示されるので、彼女はこのリンクを辿

り、編集フォームに自分自身の簡潔な紹介や連絡情報を入力できます。これをセーブすると、訪問者自身の「人名録(who's who)」ページが生成されます。もし気がきく訪問者ならば、「WhosWho」のページに、このページへのリンクを加えるでしょう。

その後に訪問した人は、RecentChanges(更新履歴)ページで、以上のページが最近修正、作成されたことと、ページ上の署名を見て、コメントを付加した人を知ることができ、そして訪問者自身の紹介文を見て、Jane が何者であるか簡単に知ることができます。こうした簡単にアクセスされる関係性はすべて、Wiki とページ連結の動作原理から自然に生まれるものです。

Wiki に欠かせないのが検索機能で、それは様々なやり方で表現できます。最も一般的なのが、1 個以上のエントリフォームと検索ボタンとともに たとえば「FindPage」という名前の 検索専用ページへのリンクを色々なところに置くことです。同じく広範に利用されている機能が逆リンク検索で、「トピック」ページ上で専用の操作を行えば、そのページへのリンクを含み、特定のトピックに「属する」他のページを見つけます。WhosWho ページがその一例です。

Wiki 訪問者は、FindPage(または、それに類するページ) が概して、検索機能と、図 2-4 のようにコンテンツを検索する際に役に立つ、他の様々な Wiki リソースへのリンクも組み込んでいることにすぐに気がきます。外部検索エンジンへのリンクも、こうしたページに含まれるかもしれません。

あらゆる Wiki ページが等しく編集可能なので、この種のリソースへのリンクが、他の訪問者によって頻繁に追加されることになります。定義された検索フォームやボタンについても、どのページにも自由に追加できます。

Wiki 訪問者は、自由に便利な入り口ページを選んでブックマークするか、さらに興味深いことには、Wiki 内外のリソースへのハイパーリンクを編集し、独自に作成した Wiki ページを「ポータルページ」として利用しはじめます。こうすることで、訪問者は価値があると思うページや外部のサイトを、他の訪問者と自動的に共有することになります。

2.2.3 有用性の基準

一通りやってみて、Wiki が有用であることの基準となる「Wiki のエッセンス」を、以下のようにまとめ直すことが可能だと考えられます。

Wiki は、手早く相互リンクを行う手法に加え、単純なナビゲーションモデルを採用し、「1 クリックで行える」というコンセプトの実現を促進します(その手法のおかげで、少なくとも Wiki ページ間でリンク切れがないことも保証されます)。

ページ内容の編集は、必要なときに単純な「マークアップ」を用い、ウェブブラウザか



図2-4

PPRやWardのWikiにおける、FindPageの形態とリンク

ら文章を入力してやれば、「1クリックで行えます」。

世界中の誰でも、何でも変更可能です(またそれを元に戻すこともできます)。

Wikiは、高速な検索機能を内蔵します。ページタイトルがリンクになっているからです。

以上は、後で実証するように、非常に実用本位な「過ぎたるは及ばざるが如し (less-is-more)」、「多いほど実質は少ない (more-is-less)」という見方に置きかえられます。

このナビゲーションモデルから得られる派生的な効果には、以下のようなものがあります。

変更履歴リストを含む、たくさんの変更可能な入り口

新しい関係性を反映するページの相互リンクによる柔軟な再構築

マルチスレッド化された、非線形の議論

書きやすさ、共同作業のしやすさ

Wikiは、基本的な構造的マークアップをある程度サポートしていますが、全体としては

装飾的なスタイリングやレイアウトに関するオプションが、明らかに欠けていることに気付くでしょう。プレーンテキスト以外のものを入力するにはあまりにも原始的なユーザーインターフェイスである、ブラウザの入力フォームでコンテンツを編集させるので、それもある程度は仕方ありません。根本的に考えてみると、最も基本的なコンテンツのマークアップ以上のものを提供する意義はほとんどありません。というのも、HTMLで提供される Wiki コンテンツの大部分は、それ以上のものをまったく必要としないからです。いずれにせよ、状況に応じてさらに機能を追加するのは容易です。

Perl で書かれたオリジナルの Wiki スクリプトは、基本機能の必要条件を満たすだけのクイックハックでした。初期バージョンが抱えていた問題として、それが文字どおりデータベースの概念を採用していて、Perl の dbm データベース機能を利用していたことがありました。最も一般的で、自由に入手可能な Perl 用 dbm モジュールに著しい制限があったため、ページサイズもかなり制限されたのです。

そのうち、この初期の「Wiki バージョン 1」は、経験から得た知識とユーザーからのフィードバックを基に改良されました。その他の開発者も、Ward のオープンソース版の Wiki のソースを入手し、彼ら独自のバリエーションを作成しました。そうしたバリエーションの 1 つは、Ward 自身の Wiki にも後のバージョンで反映されたのですが、ページが独立した「均一テキスト (flat-text)」ファイルとして格納されるモデルをとっています。その場合、各 Wiki ページは、ハードディスク上のプレーンテキストのファイルに格納されたデータに相当します。より単純で制限のないページ管理を行うと、ストレージや検索に関して、わずかながら性能低下につながるようになります。

ここまでが導入部分のセクションであり、これ以降は、標準的な Wiki を利用する基本的技術と、その技術が訪問者によるコンテンツオーサリングにどのように影響を及ぼすかということについて解説していきます。

2.2.4 Wiki の原理

ユーザーの視点で見ると、Wiki の操作の流れはとても円滑です。これは、ページを読むにしろ、リンクを辿るにしろ、利用者が行える操作の数が比較的少なく、かつその操作が簡単だからです。もしまだ中身が存在しないページを指すリンクがあれば、あなたはそのリンクを辿ることで、すぐにコンテンツを入力し、ページを生成できます。このように操作の数が少なくてすむことこそ、大抵のユーザーが望んでいることなので、Wiki における操作の選択肢を制限して、インターフェイスと操作を簡潔にしているわけです。これで大部分の人が満足するでしょう。

しかも、Wiki にコンテンツを入力するために最終的に必要なのは、「書くことだけ」です。基本的に、電子メールを書くときと同様の、今では一般に広まっているルールを適用で

きるわけです。Wikiの基本的な機能性は、その約束事に合わせて作られています。WardのWikiにおいて、システムの仕組みについて書かれたページには、以下のように記述されています。

あなたはWikiWikiWebと呼ばれるプログラムが稼動するデータベースを閲覧しています。そして、そのプログラムには確固たる主張があります。プログラムは、誰もが作者になることを望んでいます。つまり、プログラムは作者を優先するので、読者にとっては不便な場合もあります。

Wikiのオーサリングは、文章のみの環境で行われ、書式上の装飾がほとんどなく、後はマークアップ規則に関するものしかありません。利用者がまず期待する、今どきのWhat You See Is What You Get(WYSIWYG)²編集とは違っているといわれるかもしれませんが、Wikiはそうした環境でオーサリングを行います。現実でのWYSIWYGの解釈は、どんな場合でもYou Get What You Got(YGWYG)の方に近く、しばしば本質的でないのに複雑で詳細なレンダリングをフォーマットしようとして、オーサリングのために貴重な時間を浪費しています。

Wikiを考える上で欠かせないものの2つめに、ハイパーリンク機能があります。面白いことに、Wikiデータベース内では、「リンク切れ」のようなものは存在しません。まだ存在しないページに対してレンダリングされるリンク形式「？」マークのリンクを挿入するのが慣習になっています。これが視覚的なヒントとなり、そこに欠けているコンテンツを提供するよう訪問者に促すのです。したがって、その「中身のない」ページを作成しようとすると、直接コンテンツの入力を行うフォームに進むので、ページ作成作業をやりかけている場合と変わりありません。そうしたリンクが、図2-5にキャプチャされた画面ですが、このWikiには著者の名前に対応するページがありません。頭に「？」を付ける方が、オリジナルのWikiや多くのクローンが採用している後ろに「？」を付けるスタイルよりも、句読点の邪魔になりにくい傾向があることに注目してください。

以下は、ページやリソースへのアクセスがどのように動作するかということについてのまとめです。

ページリンクは、できる限り簡単に作成できるよう設計されています。Wikiの規則では、「EmbeddedCapitals」形式を利用することになっています。つまり、キャピタライズ(先頭を大文字に)した単語を結合させて、一般に「WikiWords」と呼ばれるものにすることで、それをタイトルにしたページへのリンクを定義するのです。

(URLを指定して)まだ存在していないページに直接アクセスすると、そのまま編集でき、コンテンツの入力を始められる「空の」ページが表示されます。こうしたアクセスは、指定されたページへの外部リンクを辿った場合や、まだ中身のない新規のWikiを



Wiki Book Project

Notes for the Wiki Way, ?BoLeuf, ?WardCunningham

RecentChanges

- BookPresentation
- BookOutline

図 2-5

ページリンクを表示する Wiki ページの細部。? が頭に付けられたリンクが 2 つあり、それらはまだコンテンツが規定されていないページをリンクしていることに注意。いずれかのリンクを辿ると、新規コンテンツを入力する編集フォームが開き、対応するページが作成されます。

立ち上げる場合に起こります。

未定義のページへのリンクは、通常はタイトルのリンクパターンに「?」マークのハイパーリンクがくっつくという特殊な形でレンダリングされます。複数ユーザー環境や公開環境の場合、このリンクが視覚的な意味も持ち、編集フォームに導くことで、読者にそのページを完成するよう促します。また PIM スタイルの Wiki の場合は、ページ作者にターゲットページがまだ作成されていないことを合図する役割を果たします。いいかえると、コンテンツがまだ埋まっていないということです。

新規ページを作成すると、作者は 1 つ前のページからその新規ページに移動して作業を行うことになり、ページの関係性のベースとなるツリー構造が自動的に作られます。

Tip 2.4 「ページ」はサーバーやストレージの詳細を隠蔽する

Wiki ページの作者は、サーバー上にファイルを作成したり、ページをアクセス可能にする仕組みについて、まったく気にする必要がないことに注目してください。こうしたことはすべて、Wiki によって透過的に処理され、「ページ」というメタファーだけを提示しています。

「外部」リソースに関する処理は簡単です。

外部リソースへの URL (インターネットアドレス) 全体を記述するだけで、有効リンクが作成されます。そのように参照を行うのが、ページ作者にとって簡単でもあり、直

感的でもあります。このようにして、Wikiはローカルなページとインターネット上のリソースをシームレスに統合するのです。

ベースWikiは、あなたが画像ファイルのURLを記述すると、ページ内に画像を自動的に組み込みます。Wikiは、一般的な画像ファイルフォーマットへの参照を、イメージタグに変換し、ブラウザに画像をテキストと一緒にロードし、レンダリングします。他の種類のメディアでも、それらを組み込むのに必要なHTMLを生成する、適切なファイルタイプの識別コードを追加すれば、同じように処理できます。

基本的な仕組みの話はここまでにして、これ以降はより詳細な内容を扱います。ここで我々は、Wikiクローンと呼ばれる、Wikiのコンセプトを別の形で実装したものについて、手短に概要の解説を行います。Wikiの利用体験、「Wiki文化」については、第10章で詳しく考察しています。

2.3 Wikiクローン

本当に数多くのWikiのコンセプトの派生的実装が、何年にもわたり作られてきました。我々はそれらを総称して、Wikiクローン^[3]と呼んでいます。Wikiライクなアプリケーションも多数あり、これらのカテゴリーはある程度重なりますので、ユーザーの視点から見た場合、どれくらい多くの「Wiki」が、特定の相関関係を持っているのか決めるのは難しいかもしれません。

2.3.1 プログラミング言語によるWikiの実装

サーバー環境でPerlが広く受け入れられているところに、Wardによってソースがオープンに公開されたため、PerlによるWikiのハックが急速に広まることになりました。誰もがソースをダウンロードし、セットアップを行い、様々なやり方で修正を行えました。また、別の言語への移植もできました。

オープンに派生したWikiクローンには、オリジナルのWikiの存在をほのめかす名称を持つものが多く見られます。Swiki(Smalltalk Wiki、もしくはSqueak Wiki)、Pyki(Python wiki)、TWiki、Zwikiなどがそうです。それ以外にも、ここまでは関連性のはっきりとしない名前を持つものもあります。

実装に用いられる主要な言語は、現在もPerlです。これは将来的には変わる可能性があります。特にウェブサーバーにおけるPerl以外の言語のサポートがもっと一般的になり、広がっていけば変わっていくでしょう。他の言語で作成されたクローンは、未だ特定の環

境で動かすのが望ましいといえます。

表 2-1 に、実装に利用されている言語や、(容易ではありませんが)その機能の集合に基づいて、Wiki クローンをほんの少し集めてみました。代表的なクローンのグループについて、主な違いとなる特徴をいくつか指摘するために、クローンの選択を行っています。個々の実装の選択(もしくは選択しなかったこと)や取り上げる順番は、クローンに対して具体的な価値判断を行うものではありません。これらのグループのいくつかについては、表で取り上げた後でさらに解説を行います。また追加分析として、Wiki クローンの構文規則のバリエーションについて、付録 A で比較を行っています。

Perl ベースの Wiki と Squeak Wiki に関しては、本書で多く取り上げているので、表におけるコメント以上の解説が必要になります。

Perl ベースの Wiki

本書に示すソースコードは Perl で書かれており、それは著者自身の実装を基にしています。なぜ Perl なのでしょう？

Perl は文字列操作やパターンベースの検索や置換に大変強く、それこそが Wiki が行うことの大部分なのです。

Perl はフリーで、サーバー環境に必ず装備されています。実際、Perl は Linux ディストリビューションにバンドルされて出荷されており、追加パッケージをまったく必要とせず、Perl のコードを動かさず。Windows32 ビット環境への移植も可能です。Perl は、どのプラットフォームにも容易にインストールできます。多くの書籍やウェブリソースから入手可能なので、Perl は広範にサポートされています。

Apache サーバーに Perl モジュールを組みこんでコンパイルを行うことで、サーバーのオーバヘッドを減らせます。こうすることで、30 もの要素について処理が高速化されます。高速化は、トラフィック負荷の高いサイトにとって重要なことです。

欠点としては、Perl のソースは比較的簡潔でありながら、読みにくいことがあげられます。初心者にとっては、プログラムで行われていることを理解し、さらによくあるのが意図したとおりに動かない原因を理解するのが大変難しい可能性があります。

主要な Perl のリソースは、www.perl.org にあります。

Squeak Wiki

メジャーな言語による派生物といえるのが Squeak Wiki (Swiki) です。これは、Squeak が持つ、初期の「はめ込み可能なウェブサーバー (pluggable web server)」、(pws) コンセプトの自然な拡張でした。pws コンポーネントでは、ページのソーステキスト中に普通の

表 2-1 各種 Wiki クローン

Wikiの種類、クローン	使用言語、技術	解説
Wiki、多くのクローン	Perl CGI スクリプト Apache サーバー、IIS/PWS、 もしくはスタンドアロン	WardのWikiをベースにした「オリジナル」に近いタイプ。 大抵の公開 *nix サーバーにはPerlがインストール済みなので、ソースの取得やWikiの立ち上げは容易。 多種多様な実装が存在。 Perlのコードは暗号のようで分かりにくくなる可能性があります。
TWiki	Perl (拡張付き)	TWikiは企業利用向けであり、そうしたユーザーグループを対象とした多くの機能を統合しています。 それには履歴管理、テンプレート、カテゴリテーブル、変更の際の自動電子メール通知、ファイル添付などがあります。業務向けで、強力です。 主に *nix を対象としており、いくつかのUNIX ツールに依存しているため、Windows ユーザーはそうしたツールの移植版を必要とします。
Squeak Wiki、Swiki、 CoWeb	Squeak Smalltalk	Squeak VM 環境において、Squeak ウェブサーバー上で動作します。 Smalltalk 環境に統合されています。大学でよく使われているWiki。 「オープンな」OSとサーバー内のSqueakコードモジュールにアクセスするので、拡張性が高い。 Wikiページのマッピングを、ファイル名でなく数字で行います。
Python Wiki、Py Wiki、 Pyki、Pyle、MoinMoin	Python	より構造化された言語であり、多くの人にとってコードがPerlより理解しやすい。ドイツ、日本、オーストラリア、そしてニュージーランドで人気。 Pythonは、ウェブサーバーではあまりサポートされていませんが、インタプリタをインストールすることは可能。
Ruby Wiki、RWiki	Ruby	WikiWordsでなく「RD構文」を使用。Perl、Python、その他から、多くの魅力的な機能を取りこんでいる、比較的新しい言語であるRubyをベースにしています。
Zwiki	Python + Zope、XML	Zope(「Zオブジェクト公開環境」)は、共同作業に使える、オープンソースのウェブアプリケーションサーバーです(UNIXとWindowsで利用可能)。 ZopeはPythonで書かれているので、Python WikiがZopeの上で動くと、多くの強力な機能を利用できます。他にも議論用ツールが、Zope「プラグイン」として用意されています。
JOSWiki、Kehei Wiki ^[4]	Perl	

HTML構文を許容するのが簡単でもあるので、Swikiによる実装のほとんどにその機能がサポートされているのは驚くことではありません。

Squeak(公式サイトは、www.squeak.org)はSmalltalkをベースにした、全体的にオープンなアーキテクチャを持つ、プラットフォーム非依存の仮想マシン(VM)です。Squeakは、「Smalltalk-80の現代的な実装」と紹介されました。それは、デスクトップユーザーインターフェイスを考案した、Xerox PARCチームのメンバー(Alan Kay、Dan Ingalls、Ted Kaehler)を含むチームにより開発されたものです。

Squeakは民主的なライセンスを持つオープンソースであり、現行版ではSwikiコンポーネント `pws` とSwikiアプリケーションの両方を含めた環境が配布されています。これはSqueakが基本的に「すぐに使える」システムであることを意味しており、あなたのシステムに合ったVMを起動し、Squeakの内部からSwikiを起動してから、ブラウザでそれにアクセスすることになります。Swikiが稼動しているサイトのほとんどは、現在ではの後に作られたComancheサーバー上で稼動しているので、ComSwiki(Comanche用Swiki)サイトと呼ばれることもあります。

輪郭のはっきりしたインターフェイスを持つたくさんのSqueak Smalltalkコンポーネントが利用可能なので、少しの投資でシステム全体の重要なカスタマイズや修正を促進し、Swikiコミュニティは近年、かなり急激に発展してきました。

Swikiの技術は、コンピュータサイエンスの研究やクラスでの共同作業にとっても適しているので、Swikiサーバーはアカデミックな環境で広範囲に利用されています。ジョージア工科大学におけるコア開発チームの体験を基にした、もっと詳しいケーススタディは第11章で紹介します。

Swikiや、最近ではよく知られるようになったCoWebについての主なリソースは、<http://pbl.cc.gatech.edu/myswiki.1>にあります。

Ruby Wiki

プログラム言語界におけるニューカマーがRubyで(リソースはwww.ruby-lang.orgにあります)まつもとゆきひろ(Matatz)氏によって日本で開発されました。Rubyは、Perl、Smalltalk、そしてPythonといった多くの他の言語の最も優れた機能、最も強みとする機能を取り入れ、純粋な「オブジェクト指向スクリプト言語」インタプリタを目指しています。

David ThomasとAndrew Huntによる「*Programming Ruby: The Pragmatic Programmer's Guide*」(Addison-Wesleyから2000年10月に刊行。ISBN0-201-71089-7)がちょうど出版されたところです。この本は、日本語以外で書かれた初の包括的なRubyのガイド本です。我々が本書の執筆を開始してから、2つのRubyベースのWikiクローン(RWikiとTiki)が日本で登場しました。図2-6のスクリーンショットは、咳氏、なひ氏によって開発された、日本語と英語が利用可能なWiki実装であるRWikiを表示しています。

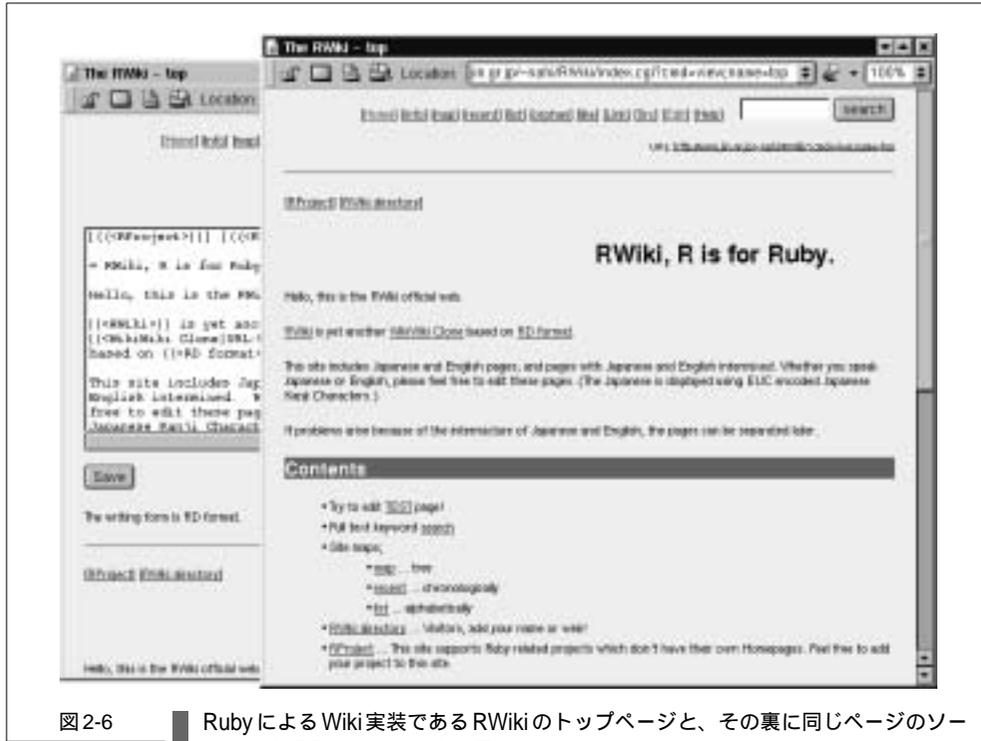


図2-6 RubyによるWiki実装であるRWikiのトップページと、その裏に同じページのソース編集を行うウィンドウを表示

RWikiは、通常のWiki構文の代わりに「RDフォーマット」マークアップを利用する点で、他のWikiと異なります。そのため、ページは任意の(ファイル名に対応する)名前を持つことが可能です。それは、日本語コンテンツを許容する際に重要な要素になります。RDは、Rubyにおける「スクリプトファイルに組み込み可能なドキュメンテーションフォーマット」で、PerlにおけるPODにやや似ています。利用は難しいものではなく、ただ他のクローンと違っていただけのことです。

このWikiクローンは新規性があり、また新しく面白い言語を使っているので、インストールやRD構文に関するドキュメントと共に、同梱CD-ROMに収録しています。QuickiWikiと同様、RWikiはスタンドアロン環境で利用できます。Rubyはまだ珍しい存在ですので、我々はWindows用に移植したバイナリと共に、最近の安定バージョン(1.6.7)のインタプリタを収録しています。RubyはGNU GPLの元で配布されます^[5]。

Rubyの作者達は、Rubyに保証されている簡潔さと能力を今後追求していくことで、Rubyがサーバー環境においてPerlにかわる強力な選択肢になるものと考えています。