

修士論文 2019 年度（令和元年度）

手書きベース Wiki システムの研究

慶應義塾大学大学院 政策・メディア研究科

早川 匠

2020 年 1 月

修士論文 2019 年度（平成元年度）

手書きベース Wiki システムの研究

論文要旨

手書きメモやイラスト等のグラフィカルなデータに自在にハイパーリンクを埋め込み、それらを Wiki として利用できる「手書きベース Wiki」システムを提案する。手書きメモやイラストは広く浸透した情報の記録・表現手法であるものの、紙を前提としたフォーマットであるために参照や管理、再利用が難しいという問題が存在する。計算機上で手書きメモの作成や管理を行うツールは広く利用されているものの、これらは紙のメモやイラストの利用形態を再現したに過ぎず、この問題を本質的に解決していない。手書きベース Wiki は、ハイパーテキスト・ハイパーリンクや Wiki 等の技術の組み合わせによって、手書きでメモやイラストを描きながら、自在にハイパーリンクを埋め込んだり、ハイパーリンクによって関連する他のメモやイラストを簡単に参照できるシステムである。既存の手書きメモ・イラストの問題点を解決するだけでなく新しい活用法を提案するため、手書きベース Wiki のプロトタイプ「DrawWiki」を実装した。本論文では手書きベース Wiki としての DrawWiki の設計や評価、応用例について述べ、研究の発展性について考察する。

キーワード

手書きメモ、イラスト、Wiki、ハイパーテキスト、ユーザーインターフェース

慶應義塾大学大学院 政策・メディア研究科

早川 匠

Abstract Of Master's Thesis Academic Year 2019

A study on drawing-based wiki systems

Summary

We propose a drawing-based note-taking style where users can use handwritten objects not only for showing shapes and texts, but for linking objects just like hyperlink texts are used for linking pages. Hypertexts are widely used on wiki systems like Wikipedia, where words and phrases are used for linking pages. Although wiki systems are useful for managing a large amount of text data, it is not possible to use non-text data for linking information. It would be more useful if handwritten drawings can also be used as hyperlinks on wiki pages just like textual phrases are used for linking pages. To prove the concept of drawing-based wiki systems, we have implemented a prototype system “DrawWiki”, where arbitrary handwritten drawings can be used as links to other pages and objects.

In this paper, we describe the design, implementation, evaluations and applications of DrawWiki, and discuss the future of wiki systems where the mixture of texts and drawings are used as hyperlinks.

Keywords

Handwritten-notes, Illustration, Wiki, Hypertext, User Interface

Graduate School of Media and Governance
Keio University

Takumi Hayakawa

目 次

第 1 章	序論	1
1.1	研究の動機	2
1.2	研究の目的	2
1.3	本論文の構成	3
第 2 章	研究背景	5
2.1	手書きのメモ・イラスト	6
2.2	タッチ・ペンインターフェースの普及	6
2.3	計算機上の手書きメモ・イラスト	6
2.4	手書きデータを表現する画像ファイルフォーマット	7
2.5	手書きデータを扱う既存のシステム	7
2.5.1	デジタルペン	7
2.5.2	メモアプリケーション	8
2.5.3	イラスト投稿・共有システム	9
2.6	テキストの進化	11
2.7	手書きデータを扱う既存システムの問題点	11
2.8	まとめ	11
第 3 章	設計	13
3.1	要件	14
3.2	ハイパーイラスト	14
3.2.1	ハイパーイラストの定義	14
3.2.2	ハイパーイラストの仕様	15
3.3	手書きベース Wiki	15
3.3.1	手書きベース Wiki の定義	15
3.3.2	DrawWiki	16
3.3.3	機能と使い方	16
第 4 章	実装	19
4.1	アプリケーション構成	20
4.2	クライアントサイド	20
4.2.1	手書きデータの表現	20
4.2.2	リンク付要素の視覚的表現	20

4.3	サーバーサイド	21
4.3.1	アプリケーションサーバー	21
4.3.2	アセットサーバー	21
第 5 章	応用例	23
5.1	個人用のメモ	24
5.2	アイデアの共有	24
5.3	共同編集	24
5.4	プロトタイピング	24
5.5	Web ページとして公開する	24
5.6	まとめ	25
第 6 章	関連研究	27
6.1	主要な研究領域	28
6.2	手書きメモ支援システム	28
6.3	デザイン・プロトタイピングツール	28
6.4	手書きデータを利用したコラボレーションツール	28
第 7 章	考察	29
7.1	評価	30
7.1.1	筆者の運用経験	30
7.1.2	ユーザーインタビュー	30
7.1.3	意見・感想	30
7.1.4	問題点・要望	30
7.2	考察	31
7.2.1	設計の妥当性	31
7.2.2	解決すべき課題	31
7.2.3	手書きメモ・イラストの問題点の検証	31
第 8 章	結論	33
8.1	研究の成果	34
8.2	総括	34
	謝辞	35
	参考文献	36

図 目 次

2.1	iPad	6
2.2	Surface	6
2.3	Anoto パターン	7
2.4	専用のペン	7
2.5	iOS のメモ	8
2.6	evernote	9
2.7	googlekeep	9
2.8	現在閲覧している作品	10
2.9	閲覧中の作品に関連する作品群	10
2.10	ニコニ・コモンズによって可視化されるコンテンツ間の親子関係	10
3.1	ハイパーイラストの概念図	14
3.2	DrawWiki の初期画面	16
3.3	ハイパーイラストの作成画面	16
3.4	ハイパーリンク埋め込み機能の操作画面	17
3.5	関連イラストの表示機能	17
3.6	関連イラストのモータル	17
3.7	エクスポート機能の操作画面	18
3.8	プリセットを用いて描いた図	18
4.1	アプリケーションの構成	20
4.2	PointerEvent API の概要	21

表 目 次

第1章 序論

本章では本研究の動機と目的、および本論文の構成について述べる。

1.1 研究の動機

手書きでメモを取ったり絵や図を描いたりすることは、情報を記録し、表現する手段として一般的であるが、その様式は紙や鉛筆等の筆記具が発明された頃からほとんど変わっておらず、一枚の紙の上で表現する事を前提としているため参照や管理、再利用が難しいという問題点が存在する。

また計算機が普及した現在では、手書きメモ・イラストを紙の上ではなくデータとして作成・記録し、それらを閲覧・管理できるソフトウェアも広く利用されているが、そのデータは JPEG¹や PNG²のような一枚の絵をピクセルの集合で単純に置き換えた形式で作成・記録されることが一般的で、紙に描かれたものと比較して本質的に変化していない。そのため、データのタイムスタンプを元に時系列で管理したり、あるいは事前に取り決めた場所 (ディレクトリ) に保存する等の工夫が必要とされ、紙の上で手書きのメモを取っていた時と同質の不便さを引き継いでいる。

一方で手書きメモやイラストと同じく紙の上で記録されていたテキストは、計算機の登場により以下のように変化した。

- 他の文書への参照を実現するハイパーリンクと、それを内包した文書の形式ハイパーテキストが登場した
- Web によって様々なハイパーテキストに手軽にアクセスできるようになった
- コラボレーションツールである Wiki が複数人による共同編集を可能にし、知見の共有を実現した

これにより参照や管理・再利用が難しいという問題が解決された。かつては手書きメモ・イラストと同様の問題を抱えていたテキストは、計算機による新しい活用法が発明された事で広く普及するに至った。そのため手書きメモ・イラストも、計算機を活用する事で同様の進化が可能であると考えられる。

1.2 研究の目的

本研究では、手書きメモ・イラストを表現する既存のフォーマットが抱える問題を解決し、またハイパーテキストや Wiki の手法を取り入れることで、従来のシステムでは実現できなかった手書きデータの参照・管理・再利用環境を実現するシステム「手書きベース Wiki」の構築を目的とする。

¹<http://www.libpng.org/pub/png/>

²<https://jpeg.org/jpeg/>

1.3 本論文の構成

本論文は以下の 8 章で構成される。

第 2 章では、本研究の背景をより詳細に分析し、既存システムの問題点を整理する。

第 3 章では、本論文で提案するシステムの基本構成と使い方について述べる。

第 4 章では、本論文で提案するシステムの詳細な実装について述べる。

第 5 章では、本論文で提案するシステムによって実現可能な応用例について述べる。

第 6 章では、関連する研究を紹介し、それらの特徴や本研究との関連を述べる。

第 7 章では、筆者による運用経験やユーザーからのフィードバックをまとめ、本論文で提案するシステムの有効性と問題点について述べる。

最後に、第 8 章で本論文のまとめと結論を述べる。

第2章 研究背景

本章では手書きメモ・イラストを扱う既存のシステムの現状と、その問題点を整理する。

2.1 手書きのメモ・イラスト

手書きによってメモやイラストを表現するのは、鉛筆等の筆記具と紙さえあればすぐに記録でき、また美麗な作品を描くことを目的としなければ、特別な技量も要求されないため、情報を記録・表現する手法として広く普及している。計算機の登場によりテキスト編集支援機能が充実したため、文章のみで完結する内容であれば手書きではなくタイピングによって記録するように置き換わったが、アイデアのような文章のみでは表現しづらい構造を持った概念を表現する場合は文字と図を自在に混合させて配置できる手書きメモの方が適している。また、テキストによってメモをとる場合はキーボードのような専用のハードウェアや、それらを使いこなすタッチタイピング等の技量が必要であるという問題点があるが、手書きの場合は紙やペン等の筆記具が扱えれば良いため、ハードウェアや技能を必要とするテキスト入力と比較してより多くの人々が利用できる手段であると言える。

2.2 タッチ・ペンインターフェースの普及

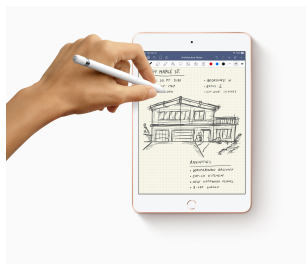


図 2.1: iPad

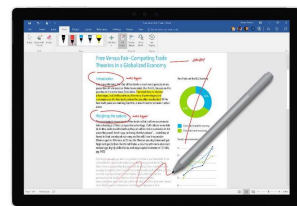


図 2.2: Surface

かつてはノートやスケッチブック等の紙の上で記録されていた手書きメモだが、タッチパネルやスタイラスペン等のインターフェースを備えたデバイスの普及に伴い、計算機上で手書きメモを取ることが一般化してきた。手書きメモやイラストを計算機の上で描く場合、マウスやトラックパッド等のポインティングデバイスではなく、ペンインターフェースが好ましいとされるが、iOS¹や Windows²、ChromeOS³等の主要なプラットフォームでスタイラスペンを備えた機種が充実しているため手書きでメモやイラストを描く環境は十分に整っている。

2.3 計算機上の手書きメモ・イラスト

計算機上でメモやイラストを作成する場合、以下のような編集支援機能を利用することができる。

¹<https://www.apple.com/jp/ios/>

²<https://www.microsoft.com/ja-jp/windows>

³<https://www.google.com/chromebook/>

- コピーやペースト
- Undo や Redo
- オブジェクトの移動や変形
- 複数のレイヤーの合成

これらの機能は紙という物理的なメディアの上では実現不可能であったが、計算機の進化によって大抵のアプリケーションに搭載されるようになったため、より便利に手書きのメモやイラストを作成することができるようになった。

2.4 手書きデータを表現する画像ファイルフォーマット

手書きのメモを作成する段階では計算機による便利な編集支援機能を利用できるようになったが、それを保存するファイルのフォーマットの機能は大きく制限されている。一般的に出力先として JPEG や PNG 等のビットマップ画像が用いられているが、この種のフォーマットはレイヤーや編集履歴等の構造は保持されず、描いたもの全てがピクセルの集合として統合・変換されるほか、一部のメタデータを除いて基本的にピクセル以外の情報を保存することができないため、紙と比較して本質的に進歩していない。そのため画像ファイルを参照するには保存するディレクトリを予め取り決めたり、タイムスタンプを頼りに時系列順でソートしたりと、紙の上で記録していたときの不便さをそのまま引き継いでしまっている。

2.5 手書きデータを扱う既存のシステム

2.5.1 デジタルペン

(1) Anoto

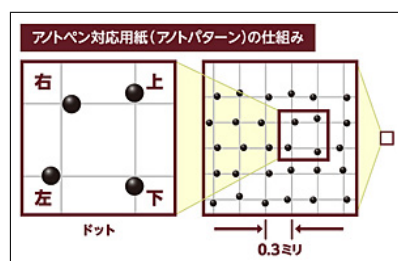


図 2.3: Anoto パターン

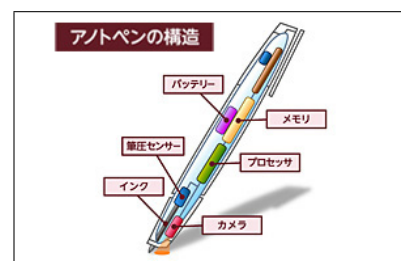


図 2.4: 専用のペン

Anoto によるデジタルペンは、Anoto パターンと呼ばれる微細なドットが印刷された専用の用紙に、カメラを内蔵したペンで記入することにより手書き入力をデータ化するシステムである。専用の紙とペン以外に特殊なタブレット機器や読み取り装置を必要としないが、あくまで物理的な紙の上に記入しているため Undo や Redo 等の計算機による編集支援機能は利用できない。

2.5.2 メモアプリケーション

手書きデータを効率よくメモとして扱えるようにしたシステムを解説する。

(1) iOS のメモ

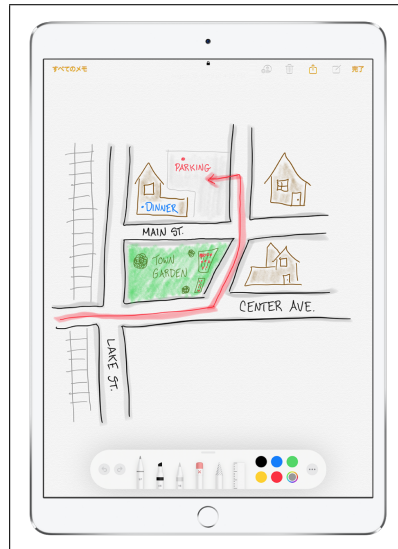


図 2.5: iOS のメモ

Apple の iPad にはメモアプリケーションが標準でインストールされており、指や Apple Pencil を用いて素早く手書きメモを取ることを可能で、さらに Undo や Redo 等の編集支援機能を備えている。一方で描いた手書きメモは画像として保存されるのみで、後から検索する手立てがなく、ファイル名の規則を決めたり保存先フォルダを分ける等の工夫が運用上必要とされる。

(2) Evernote

Evernote Corporation が開発する Evernote は指やスタイラスペンで手書きのメモやスケッチを記録することができる。手書きの文字を認識することで後からテキストによって検索する機能を備えているが、これは手書きメモをテキストに変換することで実現しているため、図形や描いたものの形状等のグラフィカルなデータから手書きメモを参照することはできない。

(3) Google Keep

Google が開発する Google Keep も、Evernote と同様にテキストによって検索することも可能である。さらに画像や手書きメモ内にある文字を認識してテキストとして抽出する機能

があるが、やはりこれも手書きメモ内の文字をテキストに変換するというアプローチであるため、Evernote と同様の問題を抱えている。

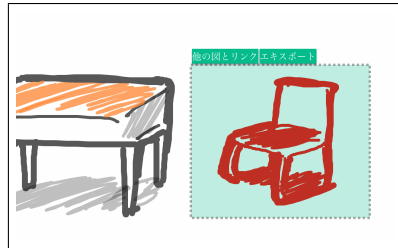


図 2.6: evernote

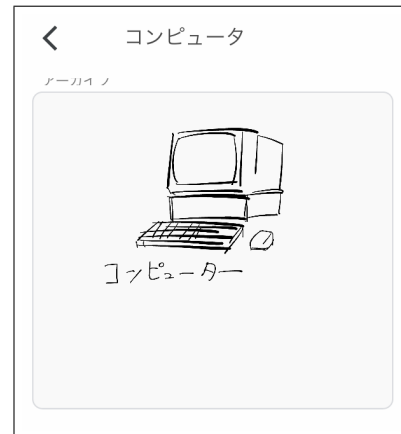


図 2.7: googlekeep

2.5.3 イラスト投稿・共有システム

Web 上で手書きのイラストを投稿・共有可能にするシステムを解説する。これらのシステムは基本的に完成されたアート作品を投稿するのが主な用途であり、メモを素早く記録する手段として使われることはないが、様々な工夫によってイラストの参照や管理の問題の改善に取り組んでいる。

(1) Pixiv

pixiv Inc. が開発する Pixiv では、投稿したイラストに複数のタグを付加することができる。また共通のタグを持つ他のイラストを関連イラストとして下部に表示する機能を備えているため、作者を横断して共通するテーマの他のイラストを参照することができる。

(2) ニコニ・コモンズ

ドワンゴが開発するニコニコ動画の関連サービスであるニコニ・コモンズでは、イラストも含めた素材の親子関係を記述するコンテンツツリーという機能が実装されている。これによりある作品の元になった作品や、ある作品を元にした他の作品を参照することができる。ただし登録は子作品の投稿者が手動で行わなければならないという制約があるため、全ての作品のコンテンツツリーが漏れなく記述されているわけではない。

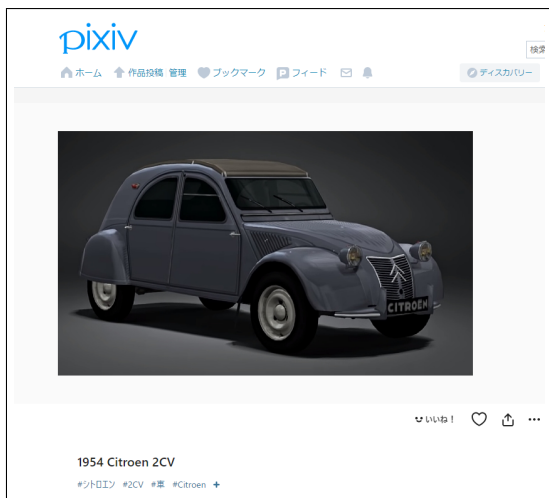


図 2.8: 現在閲覧している作品

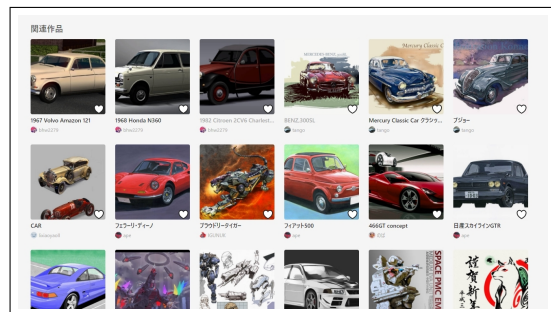


図 2.9: 閲覧中の作品に関連する作品群



図 2.10: ニコニ・コモンズによって可視化されるコンテンツ間の親子関係

2.6 テキストの進化

手書きのメモやイラストと同様に紙の上で表現されていたテキストは計算機の登場により以下のような機能を獲得した。

- ハイパーリンク
ハイパーテキストでは文章間リンクを示すハイパーリンク機能が利用でき、関連情報への素早いアクセスが可能になった。
- Web
インターネット技術の進化と Web の普及によって、世界中に存在する様々なドキュメントへ瞬時にアクセスすることが可能になった。
- 共同編集
Wiki のようなコラボレーションツールによって、場所や人数に制約を受けない共同編集が可能になった。

これにより参照・管理・再利用が難しいという問題点を解決するに至った。

2.7 手書きデータを扱う既存システムの問題点

手書きデータを扱うシステムは数多く存在し活用されているものの、画像ファイルのフォーマットの機能が制限されているため、計算機の進化によって得られる便利な機能を享受できていない。また、

2.8 まとめ

手書きメモ・イラストは広く浸透した情報の記録手法であるものの、紙というフォーマットの制約によって使い勝手が制限されている。一方でデジタル化された手書きメモ・イラストの作成や閲覧を支援するシステムが広く利用されているが、これらは計算機上で手書きの利用形態を再現したに過ぎず、本質的な問題は解決されていない。次章では上記のような問題点を解決し、これまでの手書きメモ・イラストの在り方にとらわれない次世代のフォーマット「ハイパーイラスト」と、それらを容易に管理・再利用できるシステム「手書きベース Wiki」を提案する。

第3章 設計

本章ではハイパーイラストと手書きベース Wiki の要件と設計について述べる。

3.1 要件

前章で示した画像ファイルフォーマットや手書きデータを扱う既存のツールの問題点を踏まえて、本システムの要件を整理する。

1. 簡単に手書きメモのメモやイラストが作成・編集できる
気軽に手書きメモ・イラストを取ることができ、また再編集性も可能である。
2. 作成した手書きのメモやイラストを簡単に参照したり、再利用したりできる
画像の内部に対してハイパーリンクを設定でき、関連する画像を参照することができる。

これらの要件を満たすシステムは次世代の画像ファイルフォーマットであるハイパーイラストと、その作成・編集と管理をサポートする手書きベース Wiki の組み合わせによって実現可能である。

3.2 ハイパーイラスト

本研究ではハイパーテキストの特徴を取り入れることによって既存の画像ファイルフォーマットの問題点を解決するハイパーイラストを提案する。

3.2.1 ハイパーイラストの定義

- テキストではなく手書きによって記述される
HTML 等のハイパーテキストは記法に基づいたテキストによってマークアップされるが、ハイパーイラストはタッチやスタイラスから取得される手書きデータによってその内容が記述される。
- 内部の任意の要素にハイパーリンクが埋め込むことができる
ハイパーテキストがその内部に他の文書への参照を示すハイパーリンクを埋め込むことができるように、ハイパーイラストは内部にハイパーリンクを埋め込むことができる。

図 3.1: ハイパーイラストの概念図

3.2.2 ハイパーイラストの仕様

本研究におけるハイパーイラストはSVG[1]というフォーマットをベースとしている。SVGはXML¹をベースとしており、他の画像ファイルフォーマットにはない、ハイパーイラストに適した以下のような特徴を持つ。

- グラフィカルな表現を前提に設計されている SVG には曲線等を表現する Path 要素や閉じた図形を表現する Polygon 要素等の仕様が標準で備わっている。スタイラスから得られるデータを Path 要素の属性として定義することで手書きのストロークを表現することができる。
- 構造を保持できる
ラスターイメージと異なり SVG の実体は構造化されたテキストファイルであり、線分や点はピクセルではなく独立した要素として記述される。これにより書き順等の構造も保持され、再編集性が高い。
- ハイパーリンクを埋め込める
SVG では XLink²形式のハイパーリンクを任意の要素に埋め込むことができるため画像の中の個別の要素に対して複数のリンクを定義することができる。
- Web 標準の技術である
SVG は特定の企業の製品ではなく、その仕様は全て公開されている。また作成や表示に特別なソフトウェアを必要とせず、ブラウザのみで閲覧することができる。

3.3 手書きベース Wiki

ハイパーイラストをコンテンツとする Wiki システムである手書きベース Wiki も提案する。

3.3.1 手書きベース Wiki の定義

- ハイパーイラストを作成・編集できる
タッチやスタイラスに対応したエディタを備え、手書きによってハイパーイラストを手軽に作成・編集することができる。
- ハイパーイラストにハイパーリンクを追加できる
作成したハイパーイラスト同士の相互リンクを簡単な操作で定義できる。
- 関連するハイパーイラストを参照できる
ハイパーイラスト同士のリンク関係を元に、関連するハイパーイラストを一覧できるように表示する。

この要件を満たす手書きベース Wiki のプロトタイプとして DrawWiki を開発した。

¹<https://www.w3.org/XML/>

²<https://www.w3.org/TR/xlink/>

3.3.2 DrawWiki

DrawWiki は本研究において開発した、手書きベース Wiki のコンセプトを元にしたプロトタイプとなるアプリケーションである。(図 3.2) 本章ではその主要な機能を使い方とともに解説する。

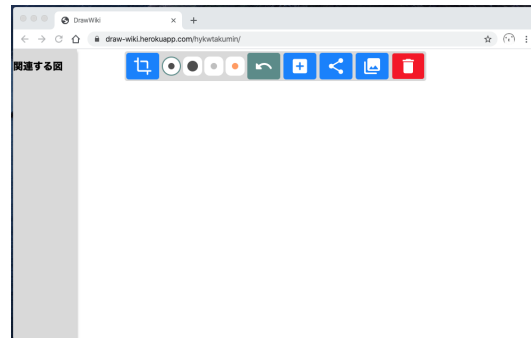


図 3.2: DrawWiki の初期画面

3.3.3 機能と使い方

(1) ハイパーリンクの作成・編集

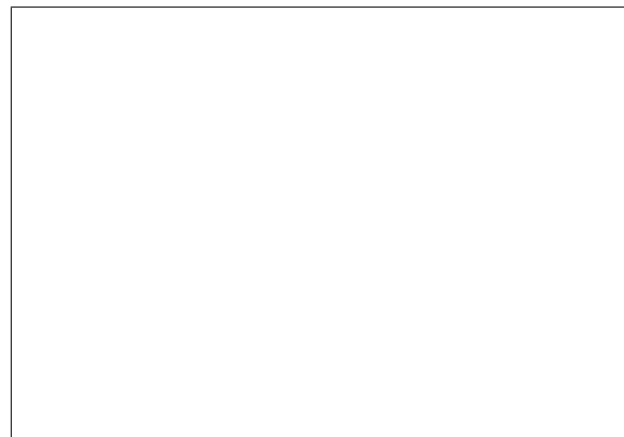


図 3.3: ハイパーイラストの作成画面

画面中央部分に自由に手書きができるキャンバスが配置されており、ここに描いたものがハイパーイラストとして自動的に保存・アップロードされる。一度アップロードしたハイパーイラストには一意な URL が割り振られるため、Web を通じて他のユーザーが閲覧し、また編集することもできる。

(2) ハイパーリンク埋め込み機能

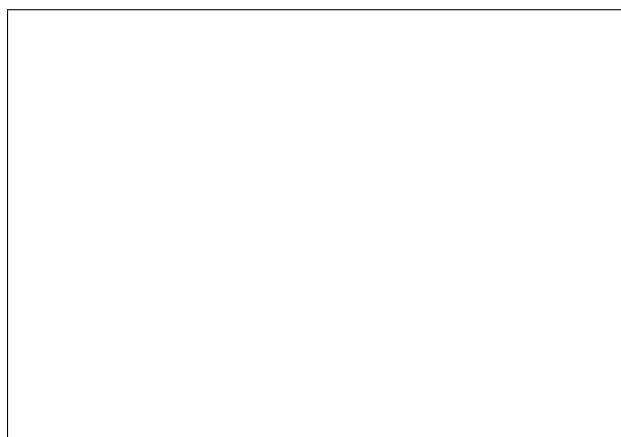


図 3.4: ハイパーリンク埋め込み機能の操作画面

範囲選択ツールを用いてハイパーリンクを埋め込みたい要素を選択することができる。要素を選択して”他の図とリンクボタン”を押すと今まで作成したハイパーイラストのリストが開き、その中からリンクさせたい図を選ぶとその図へのハイパーリンクが要素に埋め込まれる。

(3) 関連するハイパーイラストの表示機能

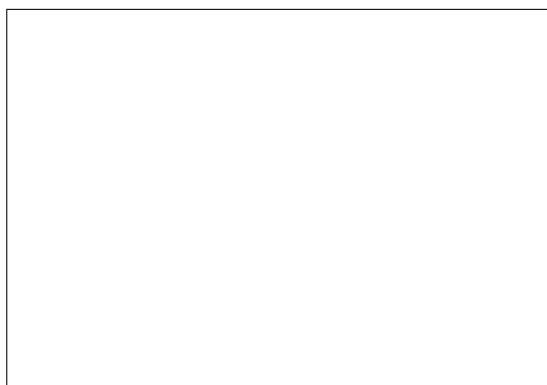


図 3.5: 関連イラストの表示機能



図 3.6: 関連イラストのモーダル

リンク機能を用いると、埋め込んだハイパーイラストのサムネイルがエディタの横の関連画像ビューに表示される。このサムネイルを選択すると、そのイラストへのリンクが埋め込まれた要素が強調表示されるとともに、そのイラストとリンクしている別のイラストをリストするモーダルが表示できる。これによりイラスト同士のリンク関係が一目瞭然に理解でき、関連イラストを参照することができる。

(4) エクスポート・共有機能

図 3.7: エクスポート機能の操作画面

DrawWiki で作成したハイパーイラストには各々に一意な URL が割り振られており、どこからでも参照することができる。また範囲選択ツールを用いて一部の範囲のみ別個のハイパーイラストとしてエクスポートすることができる。本体の SVG は

(5) ブラシプリセット

その他の特徴としてブラシのプリセットが金箱らによる Interactive Sketch[2] に基づいている。

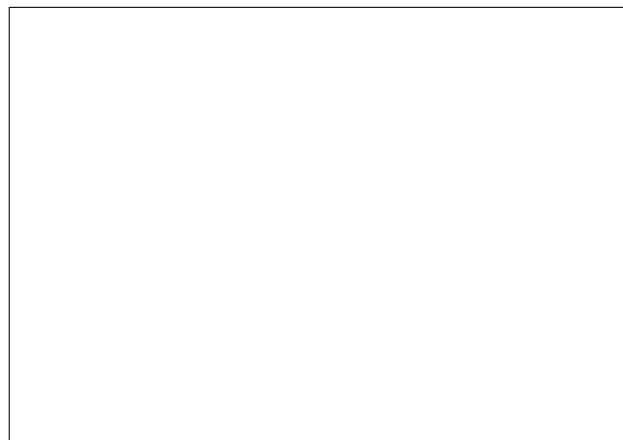


図 3.8: プリセットを用いて描いた図

第4章 実装

本章では第3章で述べたシステムの設計を受け、DrawWikiの実装について述べる。

4.1 アプリケーション構成

DrawWiki はブラウザ上から利用できる Web アプリケーションとして実装されているため、HTML5 と SVG1.1 に準拠したブラウザがインストールされていれば、OS やデバイスに依存せず利用することができる。本アプリケーションの構成は以下の図の通りである。

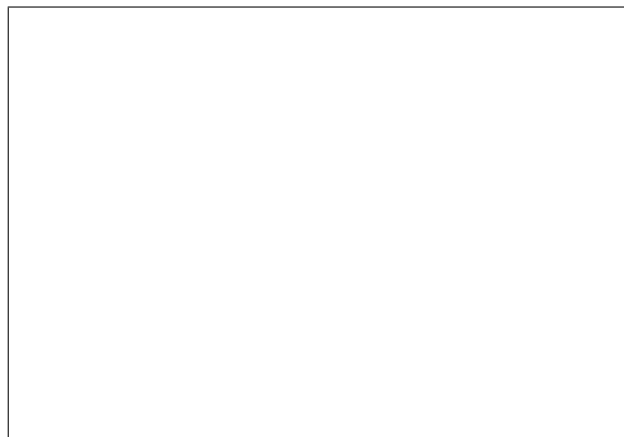


図 4.1: アプリケーションの構成

4.2 クライアントサイド

実際にハイパーイラストを作成したり関連イラストを閲覧したりするクライアントサイドのプログラムは HTML と javascript によって実装される。開発には Javascript に変換可能な漸進的型付け言語 TypeScript を、UI 操作ライブラリとして React をそれぞれ用いている。

4.2.1 手書きデータの表現

ブラウザに実装されている PointerEvent API¹によって取得したイベントからパスのデータを生成し、描画する。この API ではマウスを含めてタッチやスタイラス等のあらゆるユーザー入力を透過的に扱うことができるためあらゆる手書きデータ入力を受け付ける本システムで採用している。この API はモダンなブラウザにはおおよそ搭載されているためプラットフォームを選ばずに利用できる。

4.2.2 リンク付要素の視覚的表現

関連イラスト表示にはリンクさせた、またはインポートしたハイパーイラストのサムネイルが表示される。そのサムネイルを選択すると、そのハイパーイラストをリンクされてい

¹<https://developer.mozilla.org/ja/docs/Web/API/PointerEvent>

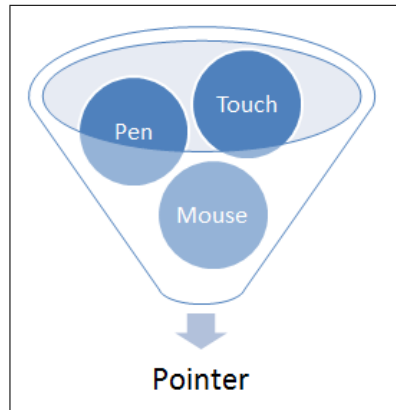


図 4.2: PointerEvent API の概要

る要素が強調表示される。SVG はスタイルシートを読み込んで適用することもできるため、CSS/CSS transition を用いてこの視覚的表現を実現している。

4.3 サーバーサイド

4.3.1 アプリケーションサーバー

DrawWiki は Node.js 上で動作する Web アプリケーションとして実装されている。HTTP リクエストを処理する Web アプリケーションフレームワークとして Express を使い、そのホスティング環境として BaaS(Backend-as-a-Service) の一つである Heroku を利用している。

クライアントサイドで生成したハイパーイラストをアセットサーバーに保存する機能を備えるほか、アセットサーバーにある既存のハイパーイラストの更新や削除等の基本的な操作を行う API を

ハイパーイラストに対してハイパーリンクを追加する処理を行った際にメタデータを更新する

4.3.2 アセットサーバー

アプリケーションサーバーはステートレスなプロセスとして実行されるためファイルとしてのハイパーイラストはアプリケーションサーバーと異なる場所にホスティングされている。クラウドストレージである AWS S3 を利用している。

第5章 応用例

本章では、手書きベース Wiki によって実現可能な応用例について述べる。

5.1 個人用のメモ

既存のツールにおいて手書きメモ・イラストは独立した画像ファイルとして扱われることが一般的であり、ファイル名の工夫、保存先ディレクトリの指定、タグ付け等の工夫を行わない限り散逸してしまい、参照や管理が面倒という問題点があった。DrawWiki においては手書きメモ・イラストそのものにハイパーリンクが含まれており、さらに関連するハイパーイラストを一覧表示することもできるため、一元管理することができる。

5.2 アイデアの共有

ハイパーイラストは

5.3 共同編集

DrawWiki で作成したハイパーイラストは URL を通じて他のユーザーもアクセス・編集することができる。このため複数人でプロトタイピングのスケッチを行う場合等、共同編集で

5.4 プロトタイピング

アプリケーションや Web ページの UI を設計する際、画面遷移を伴うモックアップを制作することがある。

5.5 Web ページとして公開する

ハイパーイラストはそれ自体がハイパーリンクを内包したハイパーテキストである。また Draw Wiki ではハイパーイラストに URL を割り振り、Web を通じてアクセス可能なようホスティングを行っている。そのため手書きという簡単な操作のみで Web ページを作り、公開することができる。

(1) 手書きベース Wiki の優位性

テキストベースの Wiki はテキストによって内容を記述することを前提としているが、テキスト入力に起因する以下のような問題点がある。

1. 効率的なテキスト入力はキーボード等の専用ハードウェアが必要であり、これらのデバイスを備えていないスマートフォンやタブレット等のハードウェアで便利に入力することができない

2. キーボードの操作にはタッチタイピング等の技術に習熟している必要があるため、利用者にはある程度の技能が要求されてしまう。
3. Wiki コンテンツの記述には専用の記法に倣う必要がある。例えばハイパーテキストを記述する上で代表的な言語である HTML では、テキストへのハイパーリンクの埋め込みを以下のように定義する。

ソースコード 5.1: html におけるハイパーリンクの定義

```
1 <a href="https://example.com">HyperLink</a>
```

また HTML へと変換できるプレーンテキストのフォーマットとして広く普及している Markdown では、同様の構造を以下のように記述する。

ソースコード 5.2: html におけるハイパーリンクの定義

```
1 [HyperLink]("https://example.com")
```

上記の記法とは別に独自の記法を採用している Wiki も存在するため、その各々の記法も利用者は網羅していなければならない。

(1) についてはタッチパネルさえあれば手書き入力が可能であるため、スマートフォンやタブレット等のキーボードを持たないデバイスからでも利用することができる。(2) に関しても、手書きはタイピングが登場する以前から存在する馴染み深いベーシックな入力方法であり、タッチタイピング等の技能を持たないが手書きはできる人でも利用することができる。ハイパーリンクの定義も作成も手書きとシンプルな操作で完結するため (3) のような記法を覚える必要もない。

5.6 まとめ

本章では、本システムによって実現可能な応用例について述べた。Wiki とハイパーリンクの組み合わせによって、既存の手書きメモ・イラストの問題点を解決することができた他、テキストベース Wiki に対しても優れた点があることがわかった。本章で述べた応用例に限らず、様々な応用が可能と考えられる。

第6章 関連研究

本章では関連研究を紹介し、それらの特徴や本研究との関連性について示す。

本研究では手書きデータを Wiki システムのコンテンツとして活用することを検討したが、手書きデータの活用を目的とする先行研究は Wiki 以外にも数多く存在する。

6.1 主要な研究領域

手書きメモ・イラストに関する主要な研究領域を解説する。

6.2 手書きメモ支援システム

6.3 デザイン・プロトタイピングツール

6.4 手書きデータを利用したコラボレーションツール

1. 手書きメモ支援システム

楽譜上のどの部分を演奏しているのか認識するためのスコアアラインメント技術が研究されている [?][?][?]。応用例として計算機による伴奏システム [?][?]、音楽鑑賞サポートシステム [?]、楽器練習サポートシステムなどが挙げられる [?]

2. 手書きデータから高度な

ジェスチャーを利用した楽譜入力インターフェースが提案されている [?][?][?]。これにより、計算機上でペンやポインティングデバイスを使って、楽譜を手書き入力することができる。

3. 楽譜認識/再生システム

画像から楽譜情報を読み取るための楽譜認識技術が研究されている [?][?][?]。これを応用して、カメラを搭載したデバイスによって印刷された楽譜を再生するシステムが提案されている [?][?]

第7章 考察

本章では、手書きベース Wiki の自身の運用経験や利用者の意見をまとめ、諸問題や研究の重要性・発展性について述べる。

7.1 評価

手書きベース Wiki のプロトタイプとなる「DrawWiki」は 2019 年 10 月に Web アプリケーションとして公開した¹。また ORF2019 において展示発表を行った。² 本節では、DrawWiki の筆者の運用経験やユーザーからのフィードバックをまとめる。

7.1.1 筆者の運用経験

筆者の運用経験

個人のメモとして

手書きの Wiki として

7.1.2 ユーザーインタビュー

7.1.3 意見・感想

授業の資料として

関連資料をまとめる手段として デザインの課題では多数のスケッチを描くことがあるが、それらは紙で管理されているため散逸してしまう (紙だと参照しづらいという問題がまず解決した) (Wiki という手法によってナレッジ化ができるようになった) (そもそも Wiki を個人で使ったことはなかったが、それはとっつきにくかったから) (手書きというカジュアルな方法で記述できるようになったので使えるようになった)

7.1.4 問題点・要望

以下のような問題点が明らかになった。

1. IME がほしい

他のメモへのハイパーリンクを埋め込む際に、どのメモを埋め込むかはモーダルによって選ぶ

2. テキスト検索によってメモを参照したい

メモの内部にある手書きの文字を対象に、テキスト検索によって参照できる機能がほしいという要望があった。現状の DrawWiki において文字として描かれた手書きスト

¹<https://draw-wiki.herokuapp.com/>

²<https://orf.sfc.keio.ac.jp/2019/>

ロックを特別に区別・認識していないため、テキストによって検索する機能を備えていない。テキストによって

3. 編集履歴が分かりづらい

基本的にハイパーイラストは URL を通じていつでも、誰でも編集可能であるが、どのように編集されたのか、どのような内容が追記されたのかを判断しづらい場合がある。また手書きストロークが誰によって追加されたのかを視覚的に表示する機能を備えていないため、共同編集を行った際に誰による編集なのかを把握しづらいという指摘があった。

7.2 考察

7.2.1 設計の妥当性

手書きメモ・イラストを相互参照可能にする

7.2.2 解決すべき課題

1. 手書き IME によるリンク追加処理の支援

描画中の手書きストロークをクエリとして検索し、類似する手書きメモを”リンク追加候補”として表示する”手書き IME”のような編集支援機能があれば、

2. テキスト検索による参照

DrawWiki ではスタイルラスから得られる手書きデータを元にストロークを生成しているため、オンライン手書き文字認識を応用することでメモ内のテキストを認識し、クエリとして検索・参照する機能は技術的に実装可能である。テキスト検索機能を備えることで、よりメモを参照しやすい状態で運用することができると考えられる。

3. 編集履歴の可視化

ハイパーイラストではストローク等の編集履歴を保持できるフォーマットである。これらの情報を元に、

7.2.3 手書きメモ・イラストの問題点の検証

第8章 結論

本章では本研究を総括する。

8.1 研究の成果

本研究では、ハイパーリンクを内蔵した次世代の手書きデータ記述形式「ハイパーイラスト」と、それをコンテンツとして扱う Wiki システム「手書きベース Wiki」の提案を行った。まず第 2 章において、既存の手書きメモ・イラストの作成・管理方法の問題点をテキストの進化と比較しながら分析した。既存システムの現状をとりあげ、計算機が普及した現在も根本的に解決されていないことを示した。第 3 章では、第 2 章で述べた手書きメモの問題点に対する有効的な解決方法を提案した。また、それに基づき本研究で開発した「ハイパーイラスト」「手書きベース Wiki」の基本構成と使い方について述べた。第 4 章では、「手書きベース Wiki」のアプリケーション構成と詳細な実装について述べた。第 5 章では、「手書きベース Wiki」によって実現可能な応用例について述べた。第 6 章では、本研究に関連する研究を紹介し、それぞれのアプローチの特徴と問題点を分析した。第 7 章では、筆者による運用経験やユーザーからのフィードバックをもとに本研究の有効性と問題点を分析した。

8.2 総括

本研究では手書きのメモやイラストの中に自在にハイパーリンクを埋め込める「ハイパーイラスト」と、それらをナレッジとして扱える「手書きベース Wiki」の開発を行った。「手書きベース Wiki」はハイパーリンクや Wiki 等の技術の組み合わせによって既存の手書きのメモ・イラスト問題点を克服するだけでなく、新しい活用法を実現した。今後は第 7 章で述べた問題点についての改善や、システムの拡張を行っていく。

謝辞

慶應義塾大学環境情報学部 増井俊之教授には学部から5年間の長きに渡りご指導を賜りました。深く感謝いたします。また、本研究の副査としてご意見、ご助言を頂きました中西泰人教授、武田圭史教授に感謝いたします。また自身の研究について幅広い議論をいただいた政策・メディア研究科博士課程の田中優氏、大和比呂志氏を初め、様々な形でアドバイスをくださった増井俊之研究会 OB 諸氏に感謝いたします。

2020 年 1 月 慶應義塾大学 政策・メディア研究科 修士 2 年 早川匠

参考文献

- [1] Dean Jackson Chris Lilley. Scalable vector graphics (svg). 10 2004.
- [2] 淳一金箱, 直蛭田, 克彦原田, 俊介高尾, 裕行佐竹, ギブソングジェームズ, 亨赤羽. 相互作用を喚起するアイデアスケッチ手法 : Interactive sketch の提案. 日本デザイン学会研究発表大会概要集, Vol. 58, pp. 14–14, 2011.