

修士論文

Web システムを用いた
商用印刷における
校正業務フローの改善

平成 22 年度修了

三重大学大学院地域イノベーション学研究科
博士前期課程

栗田新太郎

目次

第1章 はじめに	1
第2章 設計	2
2.1 一般的な印刷業務フロー	2
2.2 制作・校正状態管理の設計	3
2.2.1 制作・校正業務に関わる人と流れ	3
2.2.2 制作校正業務の状態	5
2.2.3 制作・校正業務の連絡の代替手段	7
2.3 校正機能の設計	8
2.3.1 校正の基本機能の設計	8
2.3.2 付箋アノテーションの設計	8
2.3.3 手書きアノテーションの設計	8
2.3.4 画像アノテーションの設計	9
2.3.5 複数人で校正するための設計	9
2.3.6 従来方式とWeb方式の校正方法の比較評価	10
2.4 システムの基本機能の設計	11
2.4.1 ロールの設計	11
第3章 校正システムの実装	13
3.1 サーバサイドの実装	13
3.1.1 サーバの構築	13
3.1.2 制作校正業務の状態管理の実装	14
3.1.3 デザインの分割画像を生成・呼び出しの実装	14
3.2 クライアントサイドの実装	15
3.2.1 クライアントサイド開発環境	15
3.2.2 デザイン画像の読み込みの実装	16
3.2.3 デザイン画像の移動の実装	17
3.2.4 デザイン画像の拡大・縮小の実装	18
3.2.5 手書きアノテーションの実装	19
3.2.6 付箋アノテーションの実装	20

3.2.7 画像アノテーションの実装	20
3.2.8 画面の構造	20
第4章 経緯と開発・導入	21
第5章 考察	22
5.1 印刷会社の業務の効率性について	22
5.2 顧客側のフローの改善について	23
5.3 従来の校正の代わりとなり得るか	23
第6章 まとめ	25
参考文献	25

第 1 章 はじめに

印刷会社の業務の中に校正という作業があるが、この業務は印刷業務の中での負担割合が大きく、印刷会社の業務効率向上の課題となっている。校正とは、顧客または印刷会社が制作した印刷物に文字などに誤りはないか確認を行うことである。一般的この作業は校正内容を紙ベースで顧客が確認し、それに直接手書きで加筆修正を行い、電話・FAX などを用いて校正を行っている。近年では、富士フイルム 社の XMF や Kodak 社の Prinergy などの印刷ワークフローシステムが登場した。これらのシステムを導入すると、顧客との校正業務をシステム上で効率よく行うことができるようになり、アナログでの顧客とのやりとりに掛っていた時間やコスト、ミスを削減できる。

しかし、2009 年当時から日本での普及はあまり進んでいない。その理由は、ソフトウェアの内容が日本の印刷業務フローに合致していないことが挙げられる。海外では、印刷会社の業務は顧客から送られる印刷データをそのまま印刷するだけに対し、日本では印刷会社が印刷内容を顧客と印刷会社が情報のやりとりをしながら制作することがほとんどであり、業務の内容が異なっている。そのため、研究協力した印刷会社と顧客とのやりとりにこれらのシステムをそのまま流用しても、業務フローには合致しなかった。

既存の校正システムが適合しない理由として、印刷会社と顧客の校正の進捗が分かりづらいこと、複数人での校正が行いづらいこと、校正の機能が不十分であったことが挙げられる。そこで、筆者は実際の印刷物校正業務フローに合わせ、既存の校正システムでは対応できなかったことを改善するためにシステム開発に着手した。校正業務の工程を可視化し、工程の状態管理と情報伝達手段を電子化することにより、校正業務におけるミスの削減や時間の短縮を目指した。最後に校正システム導入前後の校正回数・要した時間のデータを比較し、新しいシステムによる業務改善の効果を述べる。

第 2 章 設計

日本の印刷物校正業務に基づいた校正システム設計に関する文献がないため、校正業務とそれに対応するシステムの設計について、どのようなものが必要であり、どのように設計すればよいのかを述べる。設計されたものは Web 環境で開発を行う。

2.1 一般的な印刷業務フロー

まず、印刷会社での一般的な印刷業務フローを示す (図 2.1 一般的な印刷物業務フロー)。

1. 顧客からの受注

印刷営業が顧客から印刷物を受注する。この時、どのような印刷物にするかどうか顧客と打ち合わせ、デザインのイメージや紙の種類を決定する。

2. 原稿の受け取り (入稿)

デザインを制作する場合、必要な原稿や写真素材などを顧客からもらう。印刷営業または事務が顧客から、手渡し・メール・FAX などでも原稿を受け取る。

3. デザインの制作

顧客から受け取ったデータをデザインする。制作したデータの文字や写真に間違いがないか、印刷会社内で確認を行う。

4. 文字・デザインの確認 (校正)

出来上がったデザインを確認用の紙 (校正紙) や PDF に出力し、印刷営業または事務が顧客に手渡し・メール・FAX で送る。それを顧客側で受け取り、文字・デザインの確認を行う。これを校正と呼ぶ。

修正が発生した場合、印刷会社に差し戻し、「3. デザインの制作」に戻って修正を行う。

修正がなかった場合、次の工程に進む。これを校了と呼ぶ。

5. 版を作って印刷

印刷機で印刷できるようにデータを変換し、印刷用の版を作り、印刷を行う。

6. 配送

印刷営業や配送便で印刷物を顧客に届ける。

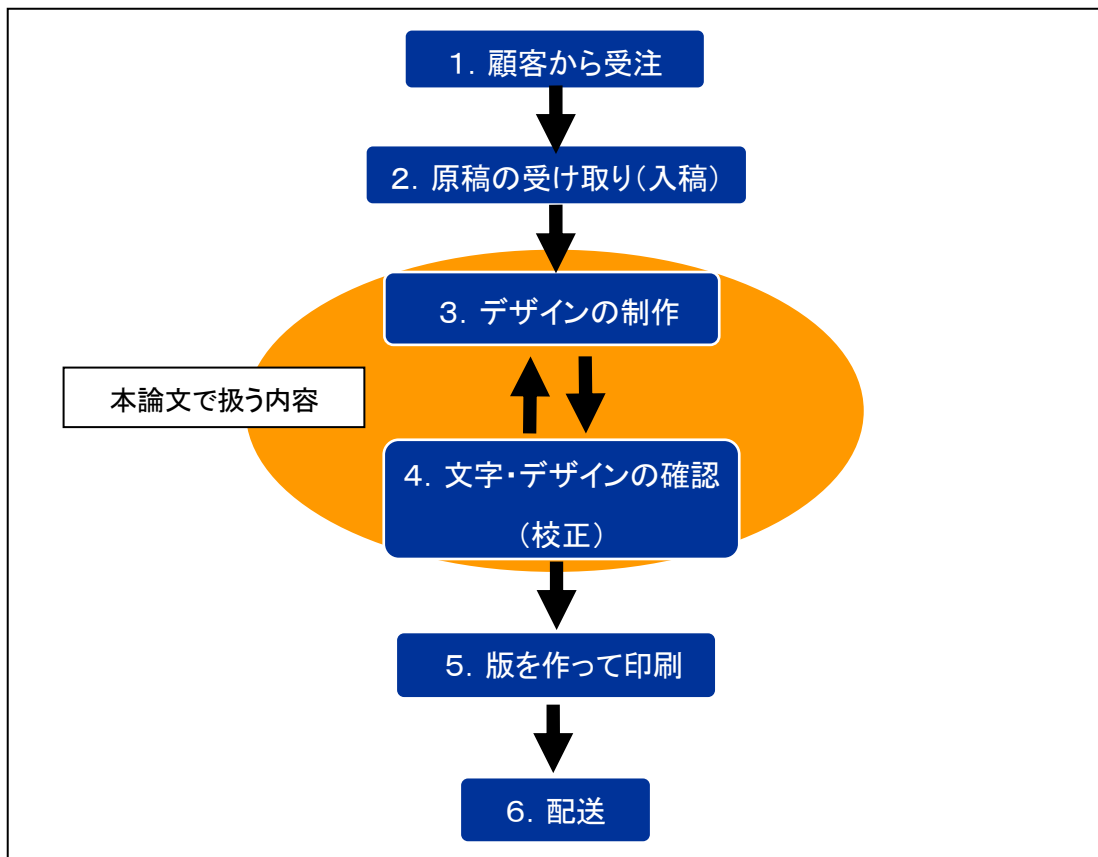


図 2.1 一般的な印刷物業務フロー

印刷業務フローの中で、特にミスが発生し、時間が掛かる部分は「3. デザインの制作」と「4. 文字・デザインの確認 (校正)」である。今後、これらをまとめて制作・校正業務と呼ぶ。顧客で修正がある度に、デザインの制作、印刷会社内での確認、顧客へのデザインの確認が繰り返されるからである。本論文では主に、この2つのシステム化、業務の改善について述べる。

2.2 制作・校正状態管理の設計

この節では制作・校正がどのように進んでいるかという状態について、視覚的に表すための設計について述べる。

2.2.1 制作・校正業務に関わる人と流れ

「3.1 一般的な印刷業務フロー」では印刷業務の流れを説明した。この節では「3. デザインの制作」と「4. 文字・デザインの確認 (校正)」(制作・校正業務)を取り上げ、

その中でどのような人が関わるかを説明する。

説明する例は、本研究において取り組んだ実際の例で示す。デザインの制作は印刷会社内で行う場合が多いが、今回は制作会社にデザインの依頼を行った（図 2.2 制作・校正業務で関わる人と流れ）。

1. 印刷会社は顧客から受注した後、顧客から受け取った原稿を制作会社に渡し、デザインの制作を依頼する。
2. 制作会社は受け取った原稿からデザインの制作を行う。
3. 印刷会社は出来上がったデザインを受け取り、顧客本部にデザインを送る。
4. 顧客本部はデザインを確認し、支店に送る。支店に送るのは複数の場合がある。
5. 顧客支店はデザインの確認をし、顧客本部に修正があるかどうか送る。
6. 顧客本部は顧客支店からの修正を取りまとめ、印刷会社に修正がない場合は校了、修正がある場合は修正の指示を行う。
7. 印刷会社は修正があった場合、制作会社に修正指示をする。
8. 制作会社は修正したデザインを印刷会社に送る。
3. にもどって校了になるまで繰り返す。

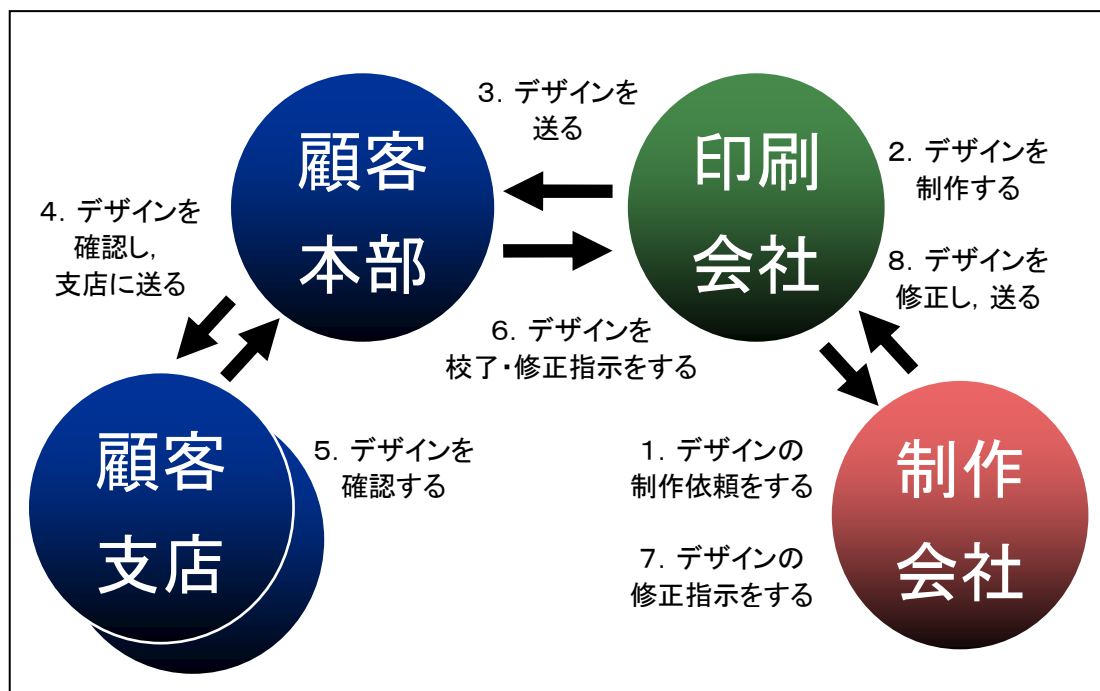


図 2.2 制作・校正業務で関わる人と流れ

以上に述べた流れを元に、校正状態を管理する仕組みを設計する。

2.2.2 制作校正業務の状態

「2.2.1 制作・校正業務に関わる人と流れ」を元に、制作・校正が今どのような状態であるかを整理し可視化した。状態は「制作・修正・校正の状態」と「顧客の校正状態」の二つに分けられた（図 2.3 制作校正業務の状態可視化のイメージ）。

- 制作・修正・校正の状態
 - 制作中・・・制作会社（印刷会社）が制作中である
 - 修正中・・・制作会社（印刷会社）が修正中である
 - 校正中・・・顧客が校正中である
 - 校了・・・顧客が校了を出している
- 顧客の校正状態
 - 未確認・・・顧客がデザインを確認していない
 - 修正あり・・・顧客が修正の指示を出している
 - 修正なし・・・顧客が修正の指示を出していない

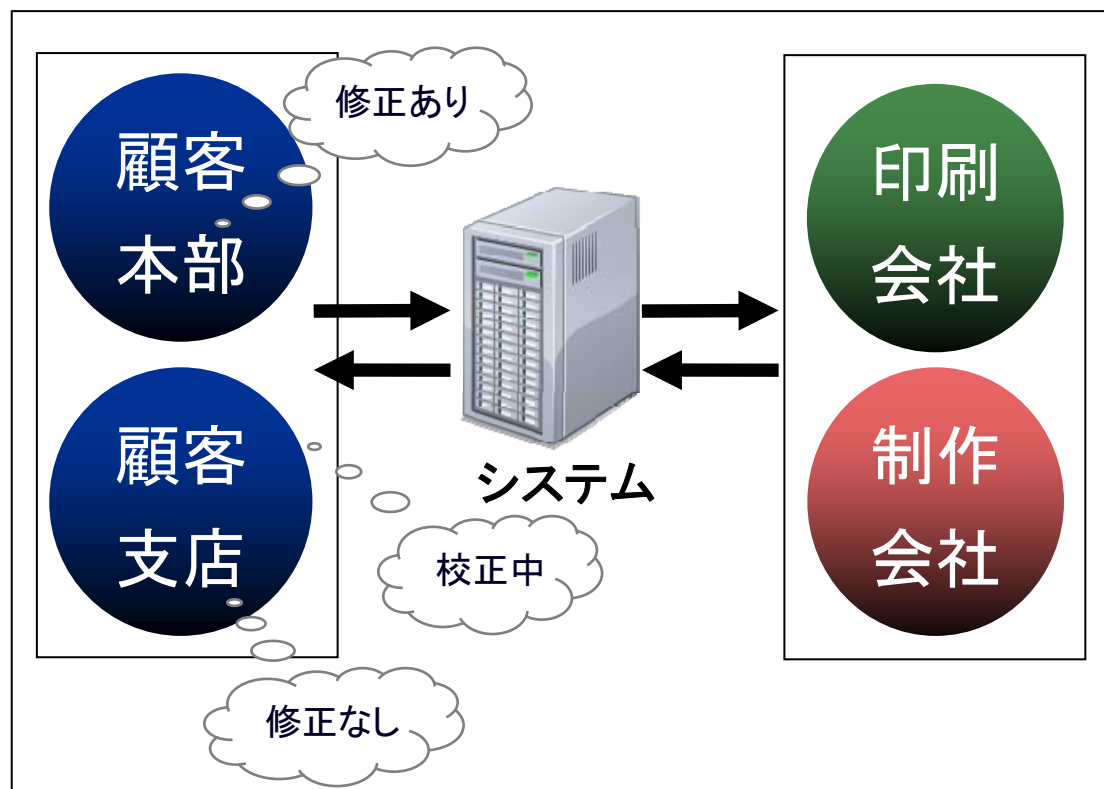


図 2.3 制作校正業務の状態可視化のイメージ

具体的に「2.2.1 制作・校正業務に関わる人と流れ」に従って、「制作・修正・校正の状態」の変化を示す（表 2.1 制作・修正・校正の状態管理）。

表 2.1 制作・修正・校正の状態管理

印刷業務フロー	制作・修正・校正
1. 印刷会社がデザインの制作依頼する	制作中
2. 制作会社がデザインを制作する	制作中
3. 印刷会社がデザインを送る	校正中
4. 顧客本部がデザインを確認し、支店に送る	校正中
5. 顧客支店がデザインを確認する	校正中
6. 顧客本部がデザインを校了・修正指示をする	修正中または校了
7. 印刷会社がデザインの修正指示をする	修正中
8. 制作会社がデザインを修正し、印刷会社にする	修正中

「制作・修正・校正の状態」を状態遷移図で表す（図 2.4 制作・修正・校正の UML 状態遷移図）。

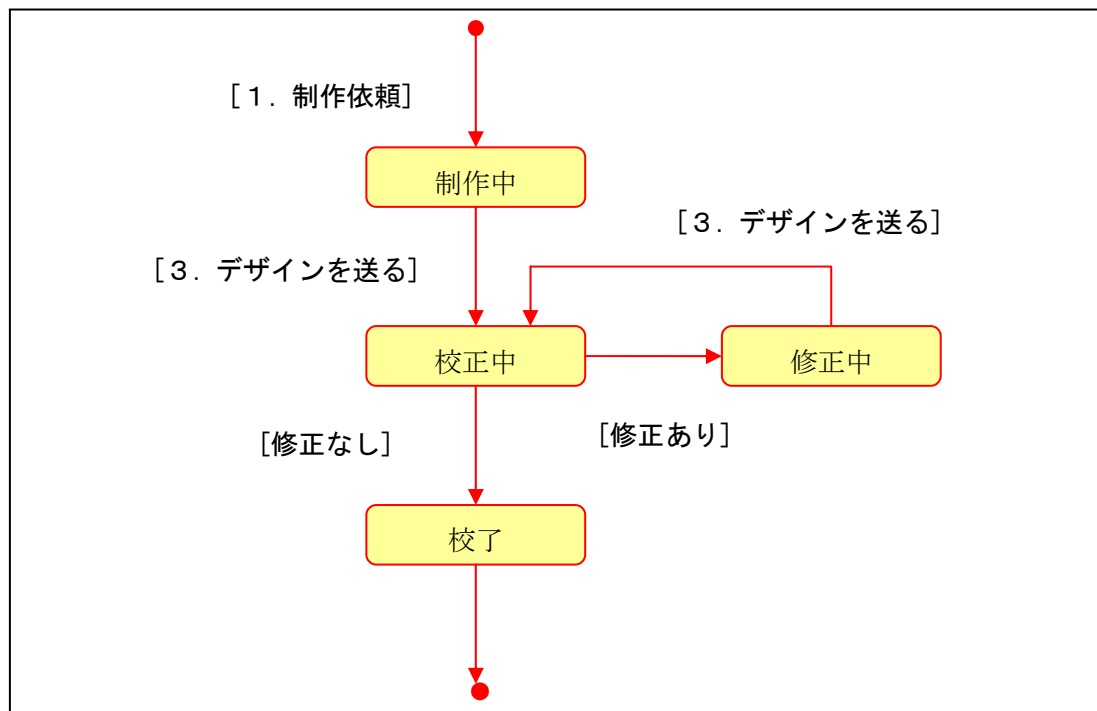


図 2.4 制作・修正・校正の UML 状態遷移図

また、「顧客の校正状態」の管理について例を示す（図 2.5 顧客の校正管理例）。

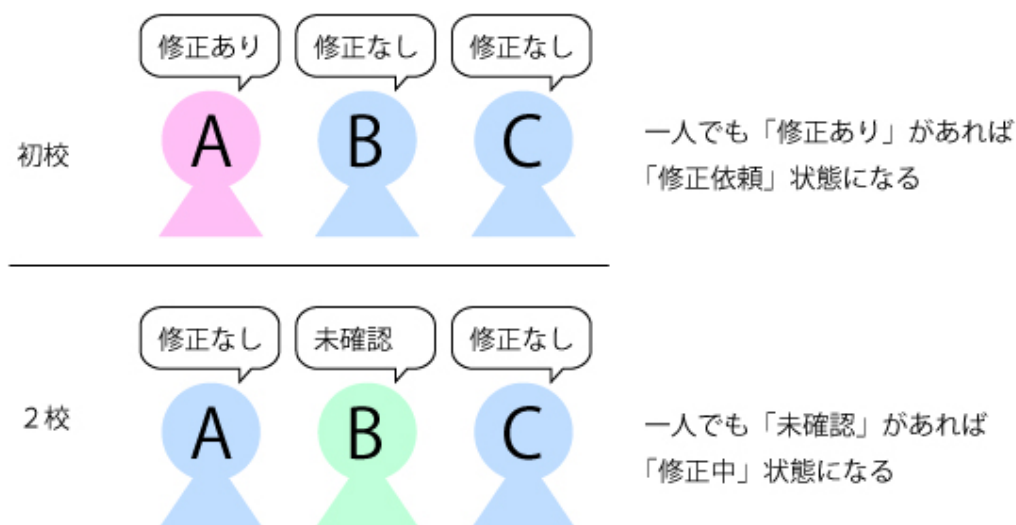


図 2.5 顧客の校正管理例

2.2.3 制作・校正業務の連絡の代替手段

制作・校正業務において、二者間での連絡は手渡し・メール・FAX で行っていた。それに代わる伝達手段を前節「2.2.2 制作・校正の状態」を用いて述べる。状態は「制作・修正・校正の状態」と「顧客の校正状態」の二種類あり、それぞれに考える必要がある。

「制作・修正・校正の状態」について二者間で連絡が必要となる場合は状態が変化した時である。状態が変化する場合は二つあり、その場合と通知先を表にまとめた（表 2.2 状態の変化と通知先）。

表 2.2 状態の変化と通知先

	通知先
3. 印刷会社がデザインを送る	顧客本部，顧客支店
6. 顧客本部がデザインを校了・修正指示をする	印刷会社，制作会社

「顧客の校正状態」についても同様に、状態が変化する時に通知が必要である。この場合は顧客本部と顧客支店の間で通知を行うようにした。

いずれの通知もメールを使って通知を行うようにした。

2.3 校正機能の設計

従来の校正作業は、一般的に紙で行い、原稿と照らし合わせて文字の間違いないかをチェックし、間違いや修正箇所があれば色つきのペンで修正の指示を入れる。それに代わる校正システムの設計について述べる。校正指示として書かれる内容は、主に、文字の修正とデザインの変更である。書かれ方として、アノテーション（注釈）の研究におけるアノテーションの付けられ方と似ている。文字の修正の指示は「付箋アノテーション」、デザインの修正の指示は「手書きアノテーション」として見なすことができる。これらを用いて校正システムの設計を行った。また、運用していくと画像を差し替える指示が多かったため、「画像アノテーション」を機能として追加した。また、複数人が校正中であると、それぞれの人の判別が難しい。これらへの対処するために、ユーザが認識できるような機能を設計した。

2.3.1 校正の基本機能の設計

紙とは異なり画面上で見る必要があるため、校正を行うときの動作を機能として置き換えなければならない。デザインを見るためには画面に表示し、細部を確認する場合はデザインを拡大して見る。全体を確認するためには、デザインの拡大・縮小や移動する必要がある。

2.3.2 付箋アノテーションの設計

文字の間違いがあった場合、紙の校正ではペンを使って校正を行っていた。それに代わる方法として、画面上の任意の場所に付箋を貼り付けられる「付箋アノテーション」を用いた（図 2.6 付箋アノテーション）。

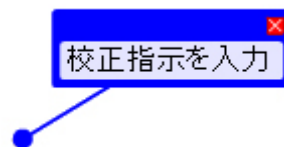


図 2.6 付箋アノテーション

2.3.3 手書きアノテーションの設計

紙の校正でデザインに修正や印を付けたい時はペンを使っていた。画面上のデザインに

も任意に校正指示を書き込むことができるように「手書きアノテーション」を用いた（図 2.7 手書きアノテーション）。

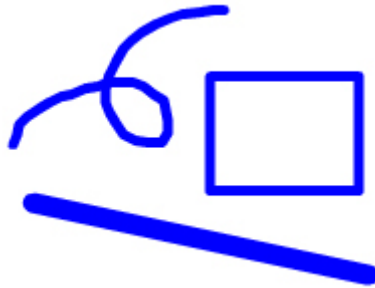


図 2.7 手書きアノテーション

また、手書きアノテーションの削除ができるように消しゴムツールを用意した。

2.3.4 画像アノテーションの設計

校正で多い指示として、写真の差し替えがある。実際の作業として、紙のデザインに修正指示を書き入れ、さらに写真を印刷会社にする必要がある。写真点数が多いと煩雑になる作業である。そこで、画面上に写真を直接差し替え指示ができるような「画像アノテーション」を用いた（図 2.8 画像アノテーション）。



図 2.8 画像アノテーション

2.3.5 複数人で校正するための設計

Web 校正システムでは複数人で校正を行うことを想定している。複数人で同時に校正を行った場合、誰がどのような校正を行ったか分かるようにしなければ、責任がどこにあるか不明になりトラブルが起こり得る。

そこで、どのユーザがどの校正の指示を行ったか、直感的に分かるように校正指示にユーザ名の表示と、色で分けるようにした（図 2.9 校正箇所のユーザ名表示と色分け）。

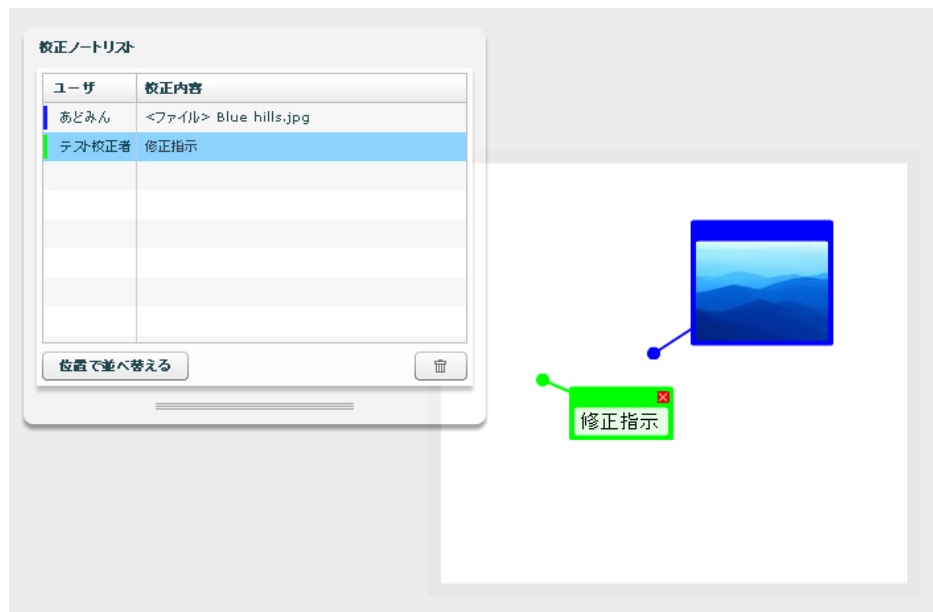


図 2.9 校正箇所のユーザ名表示と色分け

次に、これまで述べた機能を使用した例を載せる（図 2.10 校正イメージ）



図 2.10 校正イメージ

2.3.6 従来方式とWeb方式の校正方法の比較評価

校正が全て終わると印刷会社に連絡をし、校了または修正を依頼する。一連の作業を従来方式である紙での校正と Web 校正システム上で行った場合と比較した（表 2.3 紙の校

正と Web 校正システムとの比較).

表 2.3 紙の校正と Web 校正システムとの比較

校正の動作	紙での校正	評価	Web 校正システム	評価
デザインを見る	紙を見る	○	画面を見る	△
デザインの細部を見る	よく見る	○	画面上の画像を拡大して見る	◎
修正箇所を探す	目を動かして探す	○	マウスで画像を拡大・縮小や移動して探す	◎
修正箇所に印を付ける	ペンを使う	○	手書きアノテーション機能を使う	○
修正箇所に文字を書き入れる	ペンを使う	×	付箋アノテーション機能を使う	○
写真を差し替える指示をする	ペンで指示をし、写真を印刷会社にメールで送る	×	画像アノテーション機能を使う	◎
印刷会社に修正の連絡をする	印刷会社に手渡し・メール FAX などで修正原稿を渡す	×	校正完了通知をする	◎
総合評価		○		◎

2.4 システムの基本機能の設計

ここまでは校正の流れに沿って機能の設計を述べた。本節ではシステムに必要な基本機能について述べる。

2.4.1 ロールの設計

ロールとはユーザの種類に与えられた役割のことを示す。ここではロールを「図 2.2 制作・校正業務で関わる人と流れ」に示したように、「印刷会社」「制作会社」「顧客本部」「顧客支店」の 4 つのロールを定義し、限られた役割を与える。

制作会社の役割

顧客から受注があるとデザインの制作を印刷会社で行わない場合、制作会社に依頼する。

制作会社は依頼の内容を見て、デザインの制作を開始する。デザインが出来上がったら、Web システム上にアップロードし、印刷会社に通知する。

顧客の校正が終わり、修正の依頼があると通知を受け取り、システム上で校正内容を確認する。確認後、再度印刷会社に通知する。

自身が担当していない案件のものは一切見ることができない。

顧客本部の役割

印刷会社に印刷物のデザインを発注し、原稿をシステムにアップロードする。

この時、どの支店と一緒に校正を行うかの設定を行う。

その後、デザインが出来上がった通知を受け取り、校正を行う。支店からの校正完了通知を受け取り、校了か修正かを判断し、印刷会社に通知を行う。

他の顧客のものは一切見ることができない。

顧客支店の役割

顧客本部から修正の依頼の通知を受け取り、校正を行う。

自身が担当しているもの以外は見ることができない。

印刷会社の役割

顧客本部から印刷物のデザイン制作の依頼を受け取り、制作会社に制作を依頼する。

制作会社からデザインができあがった通知を受け取り、顧客本部と顧客支店に校正の依頼を通知する。

校正が終わったことを顧客本部から受け取り、修正があれば制作会社に修正を依頼する。

第 3 章 校正システムの実装

校正状態管理とアノテーション管理について、ユーザ間で情報を共有する必要があるため、Web を用いて実装を行った。また、校正画面について、アノテーションは視覚的にデザイン上に貼り付けられるようにし、アノテーションのデータは XML(データをタグ付けたフォーマット)を用いてサーバとの送受信を行うように実装した。

3.1 サーバサイドの実装

3.1.1 サーバの構築

サーバは以下のように構築した。

- OS: Ubuntu 9.04
- Web サーバ: Ruby on Rails + Mongrel
- データベース: MySQL
- 主な使用ライブラリ: ImageMagick

サーバ OS の Ubuntu は Debian をベースとした Linux ディストリビューションの一つであり、使いやすく安定していることが特徴として挙げられる。2004 年に Canonical Ltd. から支援を受けてリリースされた。サーバとしての用途も定着しており、VPS などでも Ubuntu を採択できるなど、人気が高い。本研究では使いやすさと、新しいパッケージの導入のしやすさから Ubuntu を選択した。

Ruby on Rails はプログラミング言語 Ruby をベースとした Web アプリケーションフレームワークであり、2005 年に David Heinemeier Hansson からリリースされた。多くの Web フレームワークのように、Ruby on Rails も Model-View-Controller パターンを用いている。Ruby on Rails には Web 開発に必要な基本的なツールが含まれており、“足場 (Scaffolding)”を作るもの、簡易な Web サーバ、自動タスクツールなども含まれている。DRY 原則や REST に基づいて設計されている。本研究では、高速に開発ができるため、Ruby on Rails を選択した。

3.1.2 制作校正業務の状態管理の実装

「2.2.2 制作校正業務の状態」「2.2.3 制作・校正業務の連絡の代替手段」で述べた設計を元の実装を行った。

3.1.3 デザインの分割画像を生成・呼び出しの実装

印刷で用いるデザインのデータの特性として、ベクター形式のものがよく使われる。ベクター形式とは、描画情報を線の情報として保持している形式のことをいう。本研究ではデザインを Web ブラウザ上で表示する必要があるが、Web ブラウザ上でベクター形式を表示する技術が確立されていない。表示の方法として、Adobe Flash や HTML5 の SVG など挙げられるが、印刷会社で使われているような Illustrator で作成されたデータを表示することは非常に困難である。そこで、本研究ではベクター形式ではなく、描画情報が点の集合で表されるラスター形式に一度変換して、高解像度の画像を用いることにした。変換には ImageMagick を用いた（図 3.1 ベクター形式とラスター形式の違い）。

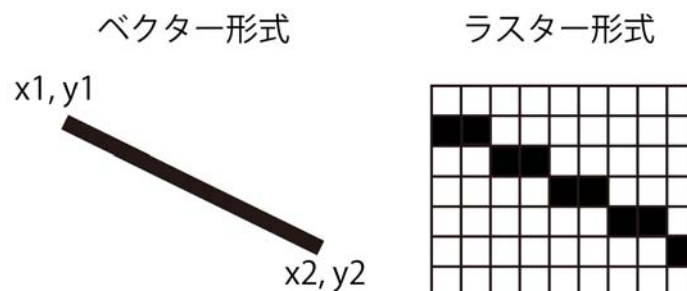


図 3.1 ベクター形式とラスター形式の違い

デザインは細部まで見られるようにするため、高解像度画像を用いなければならないが、クライアント側に読み込ませる時に高解像度のままでは通信に負荷がかかるため、画像を任意の大きさに分割する仕組みを作った。以下に画像を分割するアルゴリズムを示す。

（図 3.2 分割アルゴリズムのイメージ）。

1. デザイン画像をベクター形式からラスター形式に変換する
2. 画像の解像度を印刷用(350ppi)から Web ブラウザ表示用(72ppi)に変換する
3. 決められた分割する四辺長に合わせ、縦横の分割数を計算する。

4. 縦横の分割数で画像を順番に分割する.
5. 縮小表示用に画像を縮小し, 3 に戻って分割画像を生成する.

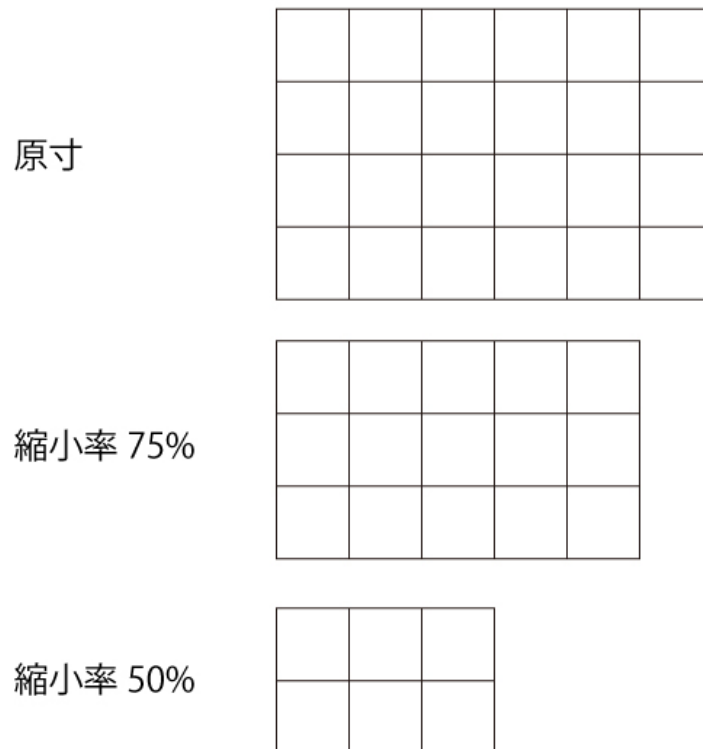


図 3.2 分割アルゴリズムのイメージ

次に分割した画像を必要な部分だけを取得する仕組みを作った. URL のパラメータに解像度(r)と分割画像を左から x 番目(x)と上から y 番目(y)を指定し, 取得するようにした.

URL のパラメータは次のようなイメージで表す.

`http://exapmle.com/proofs?r=100&x=2&y=5`

3.2 クライアントサイドの実装

3.2.1 クライアントサイド開発環境

クライアントサイドの開発環境は以下の通りである.

- 主な開発言語: Flex 3.0 (ActionScript), prototype.js
- 対応ブラウザ: FlashPlayer 10.0 以上がインストールされているブラウザ

Flex は Adobe Systems が提供している、様々なプラットフォームでリッチな Web アプリケーションを開発するための開発キットである。一般的なボタンやスライダなどのコンポーネントを利用することができる。Flex は Adobe Flash がインストールされていればどんな環境でも動作する。Flex は View を生成する xml をベースとした mxml と、静的型付け言語で Java のような記法の ActionScript を用いて記述する。以下に例を示す。

mxml の例

```
<?xml version="1.0" encoding="utf-8"?>

<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"

    xmlns:view="proofreader.views.*"

    xmlns:controller="proofreader.controllers.*">

    <controller:ProofreaderController id="proofreaderController" />

    <view:ProofreaderView id="proofreaderView" /></mx:Application>
```

ActionScript の例

```
package proofreader.controllers{

    import flash.display.*;

    public class ProofreaderController implements IMXMLObject{

    }

}
```

クライアントの開発では、画像の拡大縮小や、手書き機能などを表現するために、描画機能が強力である Flex を選択した。

3.2.2 デザイン画像の読み込みの実装

Web ブラウザ上でデザインを見るために「3.1.3 デザインの分割画像を生成・呼び出しの実装」で作成した画像を呼び出す部分を実装する部分について述べる。

デザインの画像は高解像度であるため、全てを一度に表示すると通信やメモリに負荷が掛かってしまう。そこで、必要に応じて画像を取得するように実装を行った。

分割画像は以下のように取得を行う。分割画像の画像の枚数を疑似言語で表す（図 3.2

分割画像とブラウザ表示領域の座標).

x 軸: 元デザインの左上を 0 として水平方向に軸を取る

y 軸: 元デザインの左上を 0 として垂直方向に軸を取る

w: 分割画像の四辺長

(a1, b1): 元デザインの座標を基準としたブラウザ左上の位置

(a2, b2): 元デザインの座標を基準としたブラウザ右下の位置

img: 分割画像座標. 分割画像を左上から数えた枚数を座標として表す

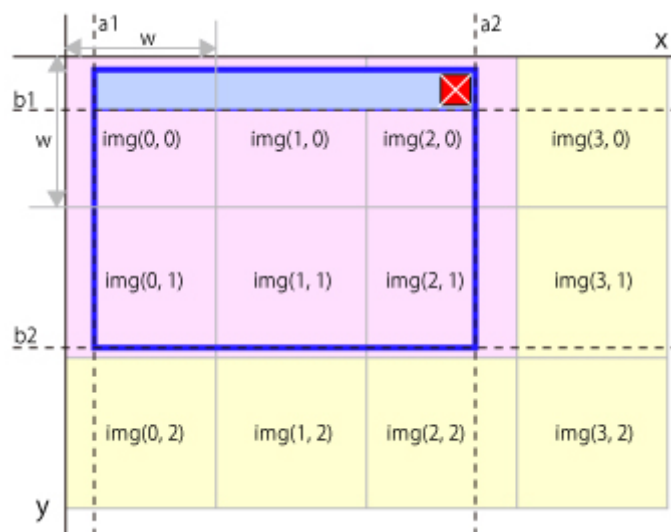


図 3.2 分割画像とブラウザ表示領域の座標

まず, 左上と右下の分割画像座標を得る.

取得する左上の分割画像座標:

$$\text{img}(x1, y1) = (\text{FLOOR}(a1/w), \text{FLOOR}(a2/w))$$

取得する右上の分割画像座標:

$$\text{img}(x2, y2) = (\text{CEIL}(a1/w), \text{CEIL}(a2/w))$$

次に, 左上と右下の分割画像座標からできる矩形内にある分割画像を順番にサーバ取り出し, 画面上に並べていく.

3.2.3 デザイン画像の移動の実装

画面を移動する時, 表示する領域も変化するため新たな分割画像を取得する必要がある.

「3.2.2 デザイン画像の読み込みの実装」と同様の仕組みで分割画像を取得する。

しかし、すでに取得している分割画像を再度読み込む必要がないため、読み込む必要があるかどうか判断しながら分割画像を読み込むようにした(図 3.3 新たな分割画像の読み込み範囲)。

必要な分割画像を読み込むアルゴリズムを簡単に示す。

1. 分割画像読み込んだ場合、その分割画像座標をキューに入れる。
2. 新たに分割画像を読み込んだ場合、キューにその座標があるかどうか検索する。
- 3.すでにその座標があった場合、分割画像は読み込まない。

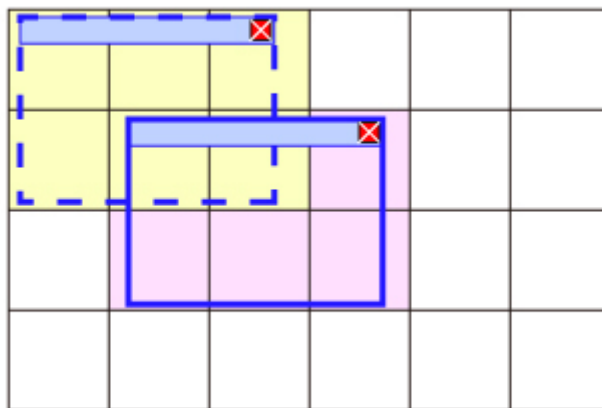


図 3.3 新たな分割画像の読み込み範囲

分割画像はキューに保存されているため、一度表示した画面を再表示させる場合には高速に行われる。また、分割画像を保存する長さはメモリによって上限があるため、必要に応じてキューから分割画像の削除を行った。これにより、従来の紙での校正で行う場合と比べ、画像をダウンロードするという欠点が改善された。

3.2.4 デザイン画像の拡大・縮小の実装

デザイン画像を拡大・縮小する場合にも新たな分割画像が必要である。新たな分割画像をサーバから読み込む場合、拡大率の情報も付けて読み込む。また、「3.2.3 デザイン画像の移動の実装」と同様にキューに保存し、一度表示した画面の再表示を高速に行うように実装を行った(図 3.4 分割画像読み込みのイメージ)。

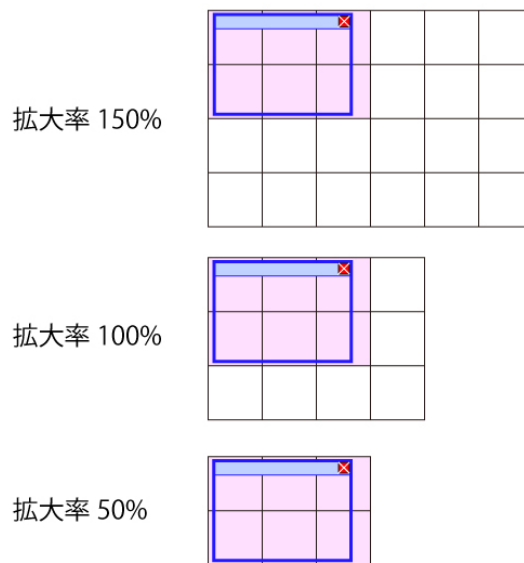


図 3.4 分割画像読み込みのイメージ

3.2.5 手書きアノテーションの実装

画面上に校正指示を直接書き込むことのできる手書きアノテーションの実装について述べる．手書きアノテーションは画面をドラッグしてなぞることでその軌跡の座標を情報として保存した．

座標の情報はそれぞれの頂点で保持するが，そのままの情報として保持すると画面が広いと座標の情報が肥大化する．そこで，情報量を抑えるために，始点から差分を取った情報を保持した（図 3.5 座標の差分）．

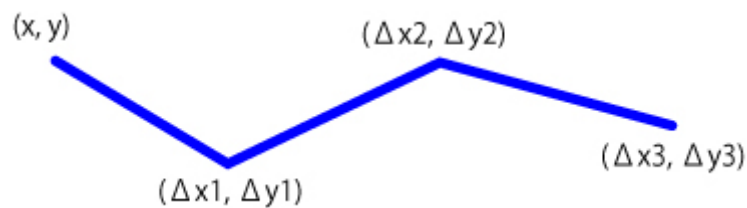


図 3.5 座標の差分

手書きアノテーションの情報はサーバと XML を介して読み込み・保存を行った．これは次節の「3.2.6 付箋アノテーションの実装」「3.2.7 画像のアノテーションの実装」でも同様に実装を行った．

3.2.6 付箋アノテーションの実装

校正画面に付箋を貼り付けることができる付箋アノテーションの実装について述べる。どの部分を校正するか、はっきりとその箇所を示すために始点を定める。次に校正の内容を表す付箋の座標とその内容を定める。付箋アノテーションはこれらの情報を保持するように実装を行った（図 3.6 付箋アノテーションの座標）。

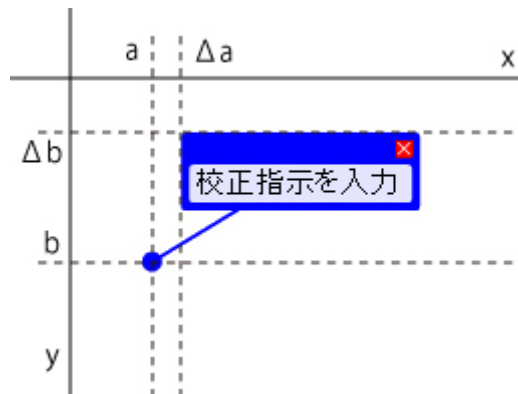


図 3.6 付箋アノテーションの座標

3.2.7 画像アノテーションの実装

画像をアップロードし、その画像を使って直接、校正指示ができる画像アノテーションの実装について述べる。画像アノテーションは付箋アノテーションの付箋部分が画像に変わったものであり、画像の情報を追加して保存した。

3.2.8 画面の構造

これまでに述べた実装を画面に並べていくと、デザイン画像、手書きアノテーション、画像アノテーション、付箋アノテーションが並び、管理がし辛い。そこで、各種類別に構造を分け、それぞれの種類ごとに管理を行った（図 3.7 画面の構造のイメージ）。

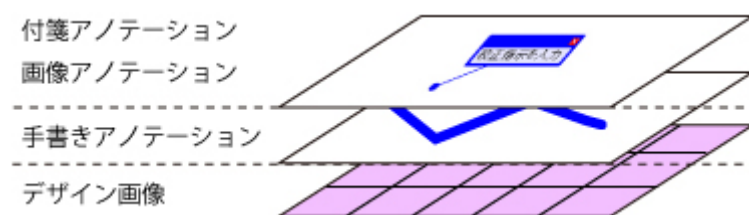


図 3.7 画面の構造のイメージ

第4章 経緯と開発・導入

筆者は印刷会社に勤務しており、2009年8月、建設・不動産業の顧客から、顧客が全国に持つ支店と印刷会社で、印刷物の校正をWeb上でできないかと持ちかけられた。2009年当時の印刷物校正のソフトウェアでは、全国各地から同時に校正を行うようなものは存在しなかったため、研究・開発することとなった。以下に開発のスケジュールを表す（表4.1 開発スケジュール）。

表 4.1 開発スケジュール

2009年8月	顧客からヒアリング
	プロトタイプ版の試作
2009年9月	顧客へプロトタイプ版導入
	顧客から評価を受け、プロトタイプを改良
2009年10月	正式版稼働
2009年11月	顧客から正式版稼働後のヒアリング
	画像を用いた校正機能の開発
2009年12月	バージョンアップ版を顧客に導入

スケジュールを進めてきたのは印刷営業1名、システム営業2名、開発1名であった。開発は少人数・短期間で進められるようにテスト駆動開発という手法で、テストを行いながら開発を行い、プロトタイプを顧客に試用してもらいながら進めてきた。システム開発自体は、印刷で使うデータをWebの画面上で見られるように工夫をすることに時間が掛かった。

正式版の導入にあたって、全国にある100支店に配布する必要があったが、本研究で開発を行ったシステムはWebを用いたシステムであったため、配布は容易であった。顧客本部と協力し、マニュアルの配布を行うことで配布を行った。

第 5 章 考察

5.1 印刷会社の業務の効率性について

校正業務で顧客とのやりとりにおいて、メールや FAX、電話などの人的な作業を減らすことができた。導入時の 2009 年 9 月のデータについて、Web システムと紙での校正を並行して業務を行っていたため、導入前と導入後の比較対象として適正だと言える。校正の回数のばらつきはあるが、次のグラフから導入後は 1.15 回減ったと言える（図 5.1 校正回数の平均の推移）。

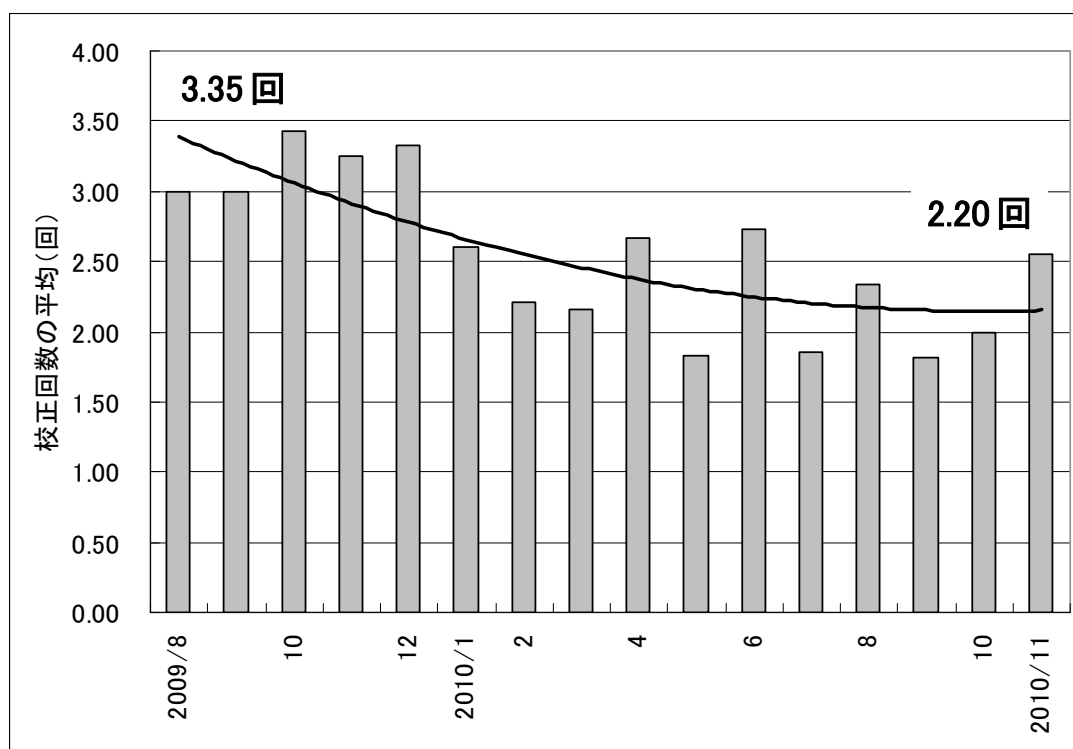


図 5.1 校正回数の平均の推移

校正回数が 1.15 回減ったことは、校正と修正、それに必要なやりとりの回数も 1.15 回減ったと言える。またそれに伴い、受注からデザインができあがるまでの日数も 7.8 日から 6.2 日となり、1.6 日減った（図 5.2 受注から校了までの平均日数の推移）。

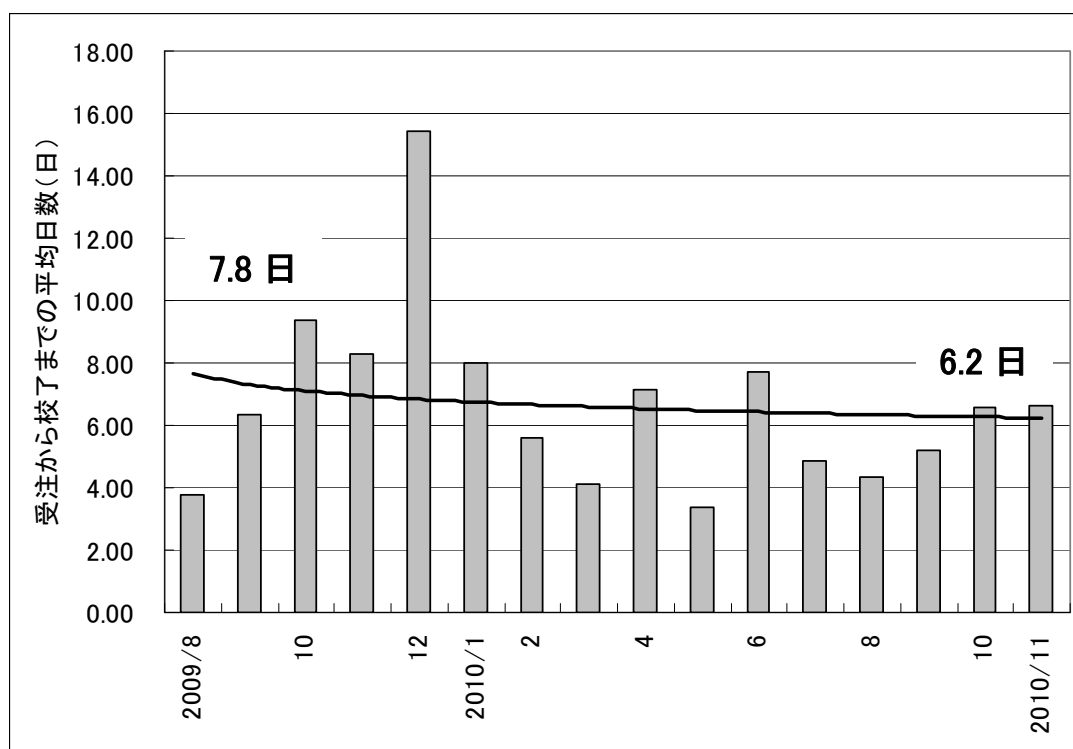


図 5.2 受注から校了までの平均日数の推移

また、顧客との校正業務で行われていた紙でのやりとりなどのアナログな作業が減り、FAX・通信費の削減ができた。

5.2 顧客側のフローの改善について

顧客はいつでもシステム上で校正を行うことができるようになり、負担が軽減された。全国の支店にも容易に導入でき、校正のコントロールができるようになったため、顧客の助けとなった。また、発注履歴や校正履歴を管理でき、印刷会社とのやりとりを明確化することができた。

5.3 従来の校正の代わりとなり得るか

従来の紙での校正に代わり、Web 校正システムが校正業務の代わりとなるかどうか、前節で述べたように校正回数や校正に掛かる日数が削減されたことで代わりになっていると言える。また、Web 校正システムはアノテーションの数が増えると見づらくなるという欠点を持っているが、運用を行った結果、1 校正あたりの付箋アノテーション・画像アノテーションの数は多くて 6 個程度であるので、画面が見づらくなってしまいうことはな

かった(図 5.3 1校正あたりの付箋アノテーション・画像アノテーションの平均の推移).

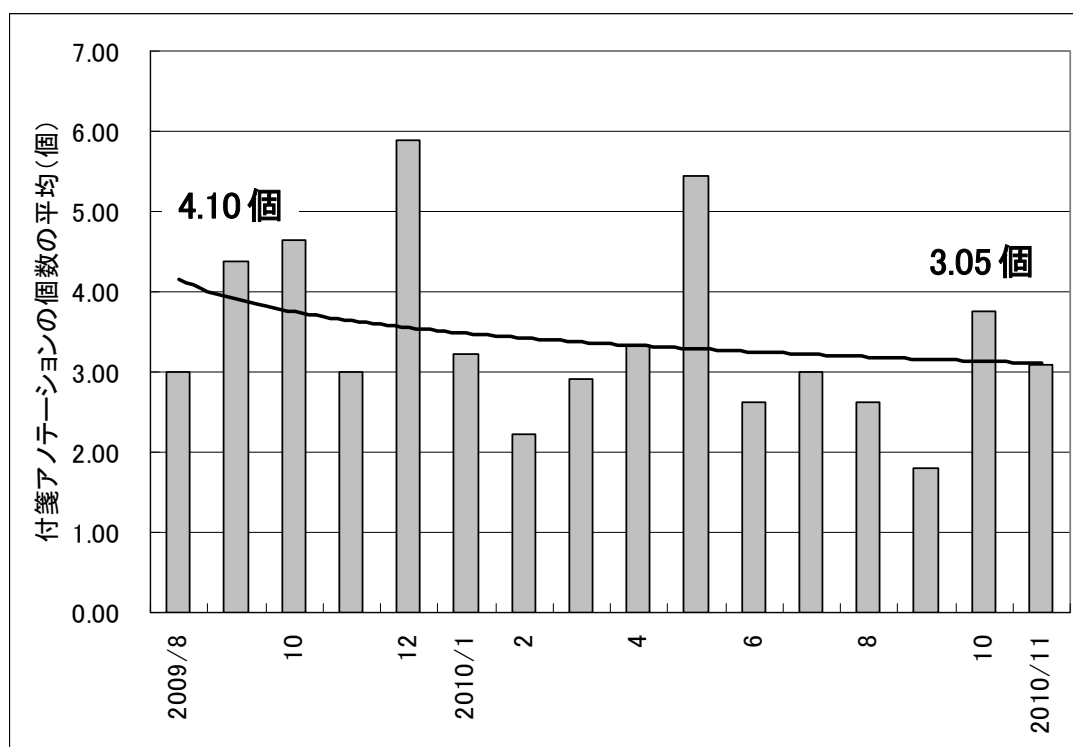


図 5.3 1校正あたりの付箋アノテーション・画像アノテーションの平均の推移

また、校正の状態が可視化され、校正箇所も履歴として残されているため、ミスの発生時の原因追及を行いやすくなった。

校正に掛かる時間・手間・利便性の全てに対して、従来の校正より改善され、校正業務の代わりになると言える。

第 6 章 まとめ

本稿では、日本の印刷業界に合致した校正システムの設計・開発についてその過程と手法を述べた。校正業務の工程を可視化することができ、従来の紙ベースでの校正に代わるシステムを設計・開発した。筆者が作成したシステムを導入する前後で、校正業務の効率が向上したことを示すことができた。よって、筆者が作成したシステムは従来の紙ベースでの校正業務の代わりになることができ、業務に掛かる時間短縮やミスの削減ができることを示した。

今回は印刷の校正業務に対して有効だと述べたが、ホームページ制作の校正業務や教育の分野などでも拡張性が高いと考えている。これからは他分野での利用を考えていきたい。

参考文献

- [1] 栗田新太郎: WEB を利用した印刷物校正システム, 第 72 回 情報処理学会全国大会 (2010)
- [2] 佐野博之, 大園忠親, 新谷虎松: 付箋アノテーションを用いた情報共有システムの試作, The22nd Annual Conference of the Japanese Society for Artificial Intelligence (2008)
- [3] 浜口拓輝, 加藤直樹, 山崎謙介: Web 上への手書きメモが共有可能なブラウザ PerowserEx の開発, 情報処理学会研究報告 Vol.2009-HCI-133 (2009)
- [4] テクニカルコミュニケーター協会事務局: PDF 電子校正向けの校正記号およびコメント入力方法のガイドライン策定, <http://www.jtca.org/standardization/index.html>, (2011/01/17)