慶應義塾大学学術情報リポジトリ
Keio Associated Repository of Academic resouces

Title	NOTA : ユーザー主導型コミュニティ活動支援システムの研究
Sub Title	NOTA : user driven approach for community activity support system
Author	永田, 周一(Nagata, Shuichi) 安村, 通晃(Yasumura, Michiaki)
Publisher	慶應義塾大学湘南藤沢学会
Publication year	2007-03
Jtitle	優秀修士論文
JaLC DOI	
Abstract	本研究は、地域コミュニティ活動で必要とされる細かな要求を満たせることができるよう、市民がソフトウェアを開発し、コミュニティで運用する中で得られるフィードバックを元に継続的に改良していく「共生開発スタイル」を提案し、事例を通じて有効性を示すことを目的としている。また、実際に共生開発スタイルの事例として開発したNOTAを利用して共生開発スタイルが広がったプロセスを明らかにしている。
Notes	安村通晃研究プロジェクト2006年
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=0302-0000-0583

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the KeiO Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

ISBN 978-4-87762-174-2
SFC-MT 2006-004

優秀修士論文

N

OTA：ユーザー主導型 コミュニティ活動支援システムの研究

2006年

永田 周一　政策・メディア研究科 修士課程

安村 通晃　研究プロジェクト

慶應義塾大学湘南藤沢学会

Keio SFC Academic Society

優秀修士論文推薦のことば

この修士論文では、「共生開発スタイル」とよぶ、地域コミュニティを念頭に置いた新たなソフトウェア開発の手法を提唱し、実際にNOTAと呼ぶソフトをこの手法により開発、展開、発展させた内容について具体的な事例を交えて説明している。これは、これまでにない新たなソフトウェア開発手法として注目されるだけではなく、実際にいくつかの地域コミュニティにおいて、実践的に使用されている点が高く評価される。

以上述べた理由により本論文を、優秀修士論文として、強く推薦する。

慶應義塾大学
環境情報学部教授
安村　通晃

NOTA: ユーザー主導型コミュニティ活動支援システムの研究

永田周一

2007 年 1 月 15 日

修士論文要旨 2006年度（平成18年度）

NOTA: ユーザー主導型コミュニティ活動支援システムの研究

近年インターネットの普及により、市民が地域コミュニティ活動のためにウェブログやSNSなどを利用する需要が高まっている。しかし、ウェブサービスや製品として開発されたソフトウェアでは、地域コミュニティで必要とされる細かな要求を満たせないという問題を抱えている。

そこで本研究は、市民がソフトウェアを開発し、コミュニティで運用する中で得られるフィードバックを元に継続的に改良していく「共生開発スタイル」を提案し、事例を通じて有効性を示すことを目的とする。共生開発スタイルは、(1) 効率をあまり考えずにシステムを継続して開発できる、(2) 利用者のフィードバックがただちに得られる、(3) 開発者とコミュニティが一体になっているため必要な機能だけを作ればよい、(4) 高いモチベーションを保って楽しく開発できる、などのメリットがある。

次に、共生開発スタイルの事例として私が開発した「NOTA」を利用して共生開発スタイルが広がったプロセスを明らかにする。NOTAは2005年より横浜市のいくつかの地域コミュニティで利用され、その後NOTAを改良したいという要望をもつ利用者の有志が集まって開発コミュニティを形成し、共生開発スタイルが始まった。NOTAは、複数の人が同時にブラウザー上で文字や図、写真などを好きな位置に書き込み、蓄積できるなど、汎用性の高いインタフェースを持つシステムである。

本研究で得られた成果は以下の通りである。共生開発スタイルによる開発を促進するために、利用者がNOTAを拡張できるプラグイン機能を追加した。この結果、NOTAを応用した地図システムや、ビデオ会議システム、障害者コミュニケーションシステムなどが地域コミュニティ内で開発された。これらの事例は、コミュニティの細かな要求に応える機能を的確かつ迅速に開発することに成功しており、共生開発スタイルの有効性を示したといえる。

キーワード：コミュニティ活動支援システム　オープンコンテンツ　共生開発スタイル

慶応義塾大学大学院 政策・メディア研究科　修士課程

永田 周一

Abstract of Master Thesis

Academic Year 2006

NOTA: User Driven Approach for Community Activity Support System

Today, through the spread of Internet, demands that citizen uses Weblog or SNS for local community activity are increasing. However, it carries problems that software developed as web services or products cannot fulfill detailed demands of local communities.

Therefore, in this study, the author proposed concepts of "civil software development" by which the citizens develop software, and improve it continuously based on feedback provided while putting it into practice in communities. Civil development has following advantages.

(1) The system can be continuously developed without concerning efficiency excessively.

(2) Feedback from users is provided immediately.

(3) They make only necessary features because developers collaborate with the community.

(4) It is possible to enjoy developing software while keeping motivations high.

Next, the author identifies developing processes of civil development in the case of "NOTA" that the author developed. NOTA have been used in some local communities in Yokohama city since 2005. Development communities were organized and civil development was adopted when volunteers of users who wished to improve NOTA gathered. With NOTA, users can put texts, figures, and photos, etc in optional positions at the same time on web browsers.

The achievements of this study are as follows: In order to promote development by civil development, the author added plug-in features that users are able to extend the usage of NOTA by themselves. As a result, extensions for NOTA were developed in local communities. A map system, a video conferencing system, and a communication system for the handicapped are some of the examples of such extensions. The effectiveness of civil development has been shown by those cases that succeeded to fulfill detailed demands of communities by archiving adequate and prompt development.

Keywords: Community Activity Support System, Open Content, Civil Software Development.

Graduate School of Media and Governance, Keio University

Shuichi NAGATA

目次

第 1 章　序論　1

1.1　研究の背景 2

1.2　研究の目的 4

1.3　論文の構成 5

第 2 章　従来の開発スタイル　7

2.1　はじめに 8

2.2　アジャイル開発手法 8

2.3　オープンソース・コミュニティ 9

2.4　従来の開発スタイルの問題点 9

2.5　共生プログラミング 10

2.6　まとめ 10

第 3 章　共生開発スタイルの提案　11

3.1　はじめに 12

3.2　共生開発スタイルの提案 12

3.2.1　共生開発スタイルの定義 12

3.2.2　共生開発スタイルの特徴 12

3.2.3　共生開発スタイルのメリット 14

3.3　NOTA ネットワーク 15

3.3.1　NOTA ネットワークの形成 15

3.3.2　NOTA ネットワークの実践 18

3.4　まとめ 18

第 4 章　NOTA の設計と開発　21

4.1　はじめに 22

4.2　従来のシステム 22

4.3　NOTA 23

4.4　NOTA の使い方 25

4.4.1　基本ツール 25

4.4.2 認証 27
4.4.3 写真の貼り付け 27
4.4.4 ファイルの貼り付け 27
4.4.5 手書き線の描画 28
4.4.6 手書き線の消去 29
4.4.7 リンク 29
4.4.8 リンクからの新規ページ作成 30
4.4.9 オブジェクトの操作と部品の重なり順 31
4.5 NOTA の実装 32
4.5.1 リアルタイム編集の実装 32
4.5.2 ユーザーアカウントと編集権限 33
4.5.3 HTML との互換性とデータ構造 34
4.5.4 手書き線の実装 34
4.5.5 機能の比較 35
4.6 まとめ 38

第 5 章 アプリケーション・プラットフォームとしての NOTA 39
5.1 はじめに 40
5.2 プラグインの使い方 40
5.3 プラットフォームの実装 40
5.3.1 NOTA API を用いた開発 42
5.3.2 NOTA API を利用した作品例 44
5.4 まとめ 45

第 6 章 NOTA の利用事例と応用 46
6.1 はじめに 47
6.2 NOTA の使い方を工夫した事例 47
6.2.1 サバイバルジュニアプロジェクト 47
6.2.2 つづき Traffic Project 47
6.3 NOTA のプラグインを開発した事例 48
6.3.1 障害者コミュニケーション支援プラグイン 48
6.3.2 NOTA ビデオ会議室プラグイン 49
6.3.3 C-Shirts プラグイン 51
6.4 利用経験 51
6.5 まとめ 54

第 7 章 関連研究 55
7.1 はじめに 56
7.2 Web 技術の新領域 56

7.2.1 非同期通信 56
7.2.2 Web の API 56
7.3 コンテンツ流通の新領域 57
7.3.1 オープンコンテンツ 57
7.3.2 Consumer Generated Media 59
7.3.3 Creative Commons 59
7.4 開発形態に関する関連研究 59
7.4.1 CSCL および CSCW 59
7.4.2 参加デザイン 60

第 8 章 共生開発スタイルの考察と今後の展望 62
8.1 共生開発スタイルの考察 63
8.1.1 オープンソースからオープンコンテンツへ 63
8.2 今後の展望 63

第 9 章 結論 65
9.1 研究の成果 66
9.2 論文の総括と結論 66

参考文献 69

付録 A NOTA ネットワークの活動履歴 72

付録 B NOTA ネットワーク運営方針 73

図目次

1.1 論文の構成 6

2.1 Waterfall Model 8

3.1 横浜市市民活動支援センターにおける NOTA の講座の様子 17
3.2 つづき人交流フェスタの活動の様子 19
3.3 NOTA バッジ 20

4.1 NOTA の編集のワークフロー 23
4.2 一般的な Wiki の編集画面 24
4.3 NOTA の協調的編集の概念図 25
4.4 NOTA の編集画面の例 26
4.5 ファイルの貼り付け 28
4.6 手書き線の描画 29
4.7 手書き線の消去 29
4.8 リンクの選択 30
4.9 NOTA による相互リンク 31
4.10 リアルタイム編集の流れ 33
4.11 表示互換性の確保 35
4.12 Flash 上で、スムーズな手書き線を描画するプロトタイプ 36
4.13 NOTA 上で、手書き線を用いてレタッチした例 36

5.1 NOTA プラグインを貼り付けた画面例 41
5.2 簡易掲示板プラグイン 41
5.3 NOTA プラグインの例 42
5.4 NOTA API マニュアル 43

6.1 サバイバルジュニアプロジェクトの作品例 48
6.2 つづき Traffic Project の表ページ 49
6.3 つづき Traffic Project の詳細ページ 50
6.4 障害者コミュニケーション支援システムの利用例 51

6.5 NOTA ビデオ会議室プラグインの利用例 52
6.6 C-Shirts の作例 53
6.7 C-Shirts の QR コードを携帯電話を利用して読み込む様子 54

7.1 Google Maps 57
7.2 Willustrator 58

表目次

3.1 コミュニティ活動支援システムの分類 14

4.1 アカウント別実行可能操作表 34
4.2 各種 Web ページ作成システムの基本機能の比較 37

第１章

序論

本章では研究の目的、論文の構成について述べる。

1.1　研究の背景

近年インターネットの普及により、市民が地域コミュニティの活動のためにウェブログやSNSなどを利用する需要が高まっている。情報技術が普及する以前から存在する地域コミュニティの活動を支援するために、Webを利用する事例が増えてきた。従来のコンピュータによるコミュニティ活動支援は、製品として開発された企業内グループウェアのようにユーザーのコンピュータリテラシーがあることを前提としたシステムや、ウェブサービスとして開発された不特定多数のエンドユーザーを対象とした仮想的なコミュニティ支援のシステムの利用が中心だった。それに対して、地域コミュニティにおけるシステムの利用は、下記のような特徴を持っている。

- 利用者のリテラシーのばらつきが比較的大きい。
- 組織としての利用目的は明確に定まっていない。
- 階層的ではなく、人的なネットワークによってゆるやかに組織化されている。
- 地域独自の利用形態に合わせて、複数のシステムを組み合わせて利用する（必要な場合は、自ら開発することもある）。
- 情報は完全に秘匿にするわけでもなく、かといって積極的に公開していくわけでもない。
- システムを開発するための予算は少ない。

上記のように地域コミュニティは、従来の利用とは異なるユーザーを対象としているものの、そのために最適化されたシステムはほとんど存在していない。既存のシステムを流用し、地域コミュニティに辛うじて適合させ利用している例が多く、コミュニティのために独自に開発を行っていることはまれである。地域コミュニティに必要なシステムの要求を定義し開発していくプロセスに、企業関係者や研究者が持続的に参加することは少なく、一時的な実験ないしは調査としてしか関わっていないように見受けられる。

筆者は、NOTAという情報共有システムを開発する過程で、2005年より横浜市都筑区において地域コミュニティにNOTAが持ち込まれ、有志により開発コミュニティが形成され、一般市民のコミュニティに普及していくという過程を経験した。NOTAの普及は、開発者（筆者）だけでは成立せず、地域コミュニティの中で利用される過程において、地域内において協力者が増え、開発と普及が同時に繰り返し進行した。NOTAには、アプリケーション・プラットフォームとして容易にシステムを拡張できる機能を実装したため、地域コミュニティの中で、NOTAを独自に発展させる者が現れた。この過程は、共著『科学技術実践のフィールドワーク』の「学習環境のデザインのためのネットワーク指向アプローチ」[上野 06b] において詳しく述べている。すなわち、システムを普及させる者（以後、「普及者」と呼ぶ）、利用する者、それらを改良する者の重層的なネットワークが構築され、開発者－普及者－利用者が一体となったコミュニティが形成された。従来のオープンソースコミュニティは、「ソースコードが共通言語」と呼ばれるように専門的な開発者に閉じたコミュニティだったが、NOTAのコミュニティは、地域コミュニティの活動と持続的な関わりを持ち、日常的な活動の一部として組み込まれながら開発が継続されている点が特徴的である。

上記のように、コミュニティ活動と持続的な関わりを持ち、開発者が市民の中から現れ、利用者と一体となって設計し開発するという視点は、従来の CSCL（Computer Supported Collaborative Learning、コンピュータに支援された協同学習）、CSCW（Computer Supported Collaborative Work、コンピュータに支援された協同作業）研究にもなかったものである。また参加デザイン（Participatory Design）のアプローチ [BG96] [KB98] [KSB98] は、ワークプレイスにおけるシステム、人工物の設計への「ユーザー」の参加をめざしたものであり、ワークショップの企画やユーザーも含めたデザイン場面を組織化するといった実践を行ってきた。しかし、こうした場面設定は一時的なものであり、長期的に関係を持続する設計者と使用者のネットワーク構築をめざすものではない。さらに、設計者と利用者は明確に区分されており、利用者が設計をサポートすることはなかった（第 7 章 関連研究で詳述）。

本研究では、以上の議論を踏まえた上で、地域におけるリアルなコミュニティに焦点を当て、私が開発した NOTA を利用して、市民がコミュニティ活動のためのソフトウェアを自ら開発する新しい開発スタイルを提示する。

1.2　研究の目的

本研究は、地域のコミュニティ活動支援システム「NOTA」を開発し、その有効性について述べる。

さらに、市民がソフトウェアを開発し、コミュニティで運用する中で得られるフィードバックを元に継続的に改良していく「共生開発スタイル」を提案する。また、2005年から形成されたNOTAの開発と普及を進めるコミュニティである「NOTAネットワーク[*1]」の事例を通じて有効性を示すことを目的とする。本研究の目的は以下のようにまとめることができる。

本研究の目的

1. 市民が地域コミュニティにおいて開発を行う「共生開発スタイル」の提案
2. コミュニティ活動支援システム「NOTA」の開発
3. NOTAを拡張するアプリケーション・プラットフォームの実現
4. NOTAの利用事例の評価と有効性の検証

1. 市民が地域コミュニティにおいて開発を行う「共生開発スタイル」の提案

   地域コミュニティにおける共生開発スタイルは、コミュニティの細かな要求を満たしたソフトウェアを開発できる。NOTAの事例に見られるような新しい開発形態を「共生開発スタイル」と名付け、提案する。また、従来のウォーターフォール・モデルやアジャイル・モデルなどの開発手法との違いを明らかにする。

2. コミュニティ活動支援システム「NOTA」の開発

   本研究では、コミュニティ活動支援のために情報共有の基盤として汎用的に用いることができるシステム「NOTA」を開発し、提案する。NOTAは、(1) 複数の人がブラウザ上でリアルタイムに情報を書き込み（協同編集機能）、(2) 情報を蓄積していつでもウェブページとして閲覧することができ（情報共有機能）、(3) 直接操作によって手書き線や写真を含めた自由なレイアウトを作成可能なシステムである。NOTAの使い方と実装について述べる。

3. NOTAを拡張するアプリケーション・プラットフォームの実現

   地域コミュニティのためのシステムは、利用者のリテラシーや活動の様態にあわせてシステムをコミュニティ内部で改変できるようにすることが重要であると考え、NOTAに、アプリケーション・プラットフォームとしての機能（NOTA API）を実装した。これによって、NOTAの利用方法を大きく変えるような変更をNOTAに加えることが可能になったが、その具体的な仕様と、実現可能な機能について述べる。

4. NOTAの利用事例の評価と有効性の検証

   実際にNOTAを使ってコミュニティ活動に応用した事例を、(1) NOTAの使い方を工夫して活用した事例と (2) NOTA自体を改良して利用した事例をそれぞれ挙げ、詳述する。また、NOTAの有効性を検証し、評価する。

[*1] ウェブサイトは、http://nota.jp/network/

1.3　論文の構成

本論文は以下のように構成される。図 1.1 に論文の構成を示す。

第 2 章　従来の開発スタイル

本研究の背景として従来の開発スタイルに関する研究について述べ、それらの問題点を指摘する。NOTA のシステムに関連する研究については、第 7 章で述べる。

第 3 章　共生開発スタイルの提案

コミュニティ活動支援のための新しい開発形態である「共生開発スタイル」を提案する。従来の開発手法と比較し、共生開発スタイルのメリットを明らかにする。本研究を進める過程において、横浜市で NOTA を利用して共生開発スタイルが広がったプロセスを説明する。

第 4 章　NOTA の設計と開発

NOTA の使い方、システム構成、実装、新規性について詳細に述べる。

第 5 章　アプリケーション・プラットフォームとしての NOTA

NOTA に実装した、アプリケーション・プラットフォームとしての機能について述べる。「NOTA API」と呼ばれる一連のインタフェースについても述べる。

第 6 章　NOTA の利用事例と応用

NOTA をコミュニティで利用した事例を紹介する。また、共生開発スタイルによって開発された NOTA のプラグインについて述べる。

第 7 章　関連研究

NOTA に関連する先行研究を紹介し、本研究との関連や差異を述べ、本研究の位置づけを明らかにする。

第 8 章　共生開発スタイルの考察と今後の展望

共生開発スタイルについて、現在分かっている課題や限界、今後の展望について述べ、本研究のさらなる発展について論述する。

第 9 章　結論

本論文を総括する。

序論

第 1 章　序論

第 2 章　従来の開発スタイル

本論

第 3 章　共生開発スタイルの提案

第 4 章　NOTA の設計と開発

第 5 章　アプリケーション・プラットフォームとしての NOTA

第 6 章　NOTA の利用事例と応用

第 7 章　関連研究

結論

第 8 章　共生開発スタイルの考察と今後の展望

第 9 章　結論

図 1.1　論文の構成

第 2 章

従来の開発スタイル

本章では本研究の背景として従来の開発スタイルに関する研究について述べ、それらの問題点を指摘する。NOTA のシステムに関連する研究については、第 7 章で述べる

2.1　はじめに

第 1 章でも触れたように、Web を利用したコミュニティ支援システムは、地域コミュニティなどの広い範囲のユーザー層にも急速に普及しつつある。エンドユーザーの細かい要求に対応するために、ソフトウェアの開発手法にも従来とは異なる視点が必要になってきている。ここでは、そうした流れの中で生まれてきた新しい開発手法のうち、特に本研究と関わりの深いものについて述べる。

2.2　アジャイル開発手法

ソフトウェアの開発手法として広く知られているのは、ウォーターフォール型と呼ばれる開発手法である。プロジェクト全体を「要求」「仕様」「分析」「設計」「プログラミング」「検査」「運用」といった工程に分割し、各工程での成果物を明確に定義し、その成果物に基づいて後工程の作業を順次行っていく（図 2.1）。後工程に移行した際に予測しない事態が明らかになり前工程に戻ることもあるが、そういった齟齬を引き起こさないために、仕様書やドキュメントを多く作成し、それらを重視する（[Rov87]）。

ウォーターフォール型は、計画的にソフトウェア開発を進めるには向いているとされるが、一方で、開発中に顧客との交渉などによって頻繁に要求が変わるプロジェクトには、不向きであると

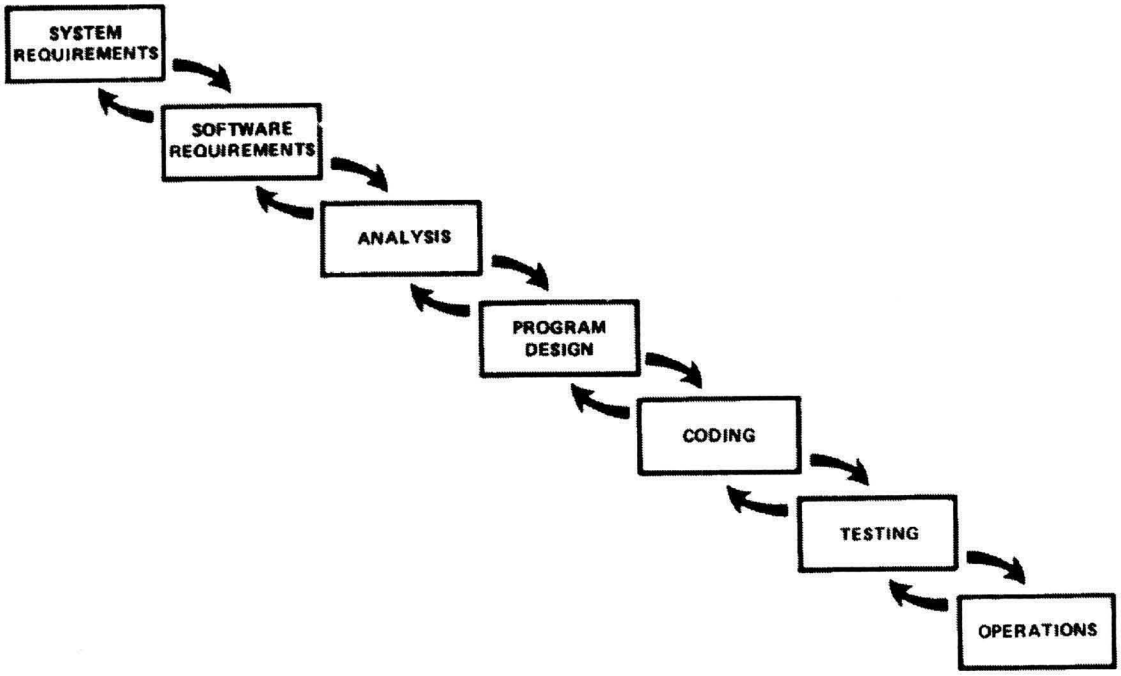

図 2.1　Waterfall Model

指摘されている。ウォーターフォール型に対して、近年 Beck[Bea] らを中心に提唱されているのがアジャイル開発手法である[*1]。アジャイル（適応型）開発手法では、最小の単位にソフトウェアを分割し、短い期間を単位としてそれらを反復的に開発し、プロジェクトを進める。各反復（イテレーションと呼ばれる）は、1 つの小規模なプロジェクトに似たようなような形態をとる。プロジェクトの全期間を通して「動いているソフトウェア」を作ることを重視する。ドキュメントや文書はウォーターフォール型に比べて少なくなる。

アジャイル開発手法は、特にエンドユーザーを対象としたウェブシステムの開発に適している。このようなシステムにおいては、顧客（エンドユーザー）が求める要求はあらかじめ明確ではない。はじめに最小限の機能を実装して試験的に公開し、ユーザーの反応を見た上で仕様を変化させながらシステムを開発していく手法は、迅速にユーザーの要求に応えることができる。

また、開発組織の形態もウォーターフォール型とアジャイル型では異なる。前者が多数の開発者がいる秩序を重視する組織文化に適しているのに対して、後者は、少ない開発者が混沌とした状況に意欲を持って取り組む組織文化に適しているとされる [BT04]。オープンソース・コミュニティは、後者の組織文化を持っていることが多く、アジャイル型が適している。

2.3　オープンソース・コミュニティ

オープンソース・コミュニティ [Sta02] は、ソフトウェアのソースコードをインターネットに公開し、世界中から有志のプログラマーが集まって仮想的なコミュニティを形成し、少しずつ開発に貢献することによってソフトウェアを開発する形態である。このようなコミュニティで開発すると、ソースコードが公開されていることによって学習が可能になったり、バグの原因を自分で確かめて修正することができたりするなどのメリットがある。「ソースコードが共通言語」といわれるように、ソースコードを理解できる開発者だけが参加するコミュニティである。

2.4　従来の開発スタイルの問題点

ウォーターフォール型とアジャイル型は、どちらも企業の工業的なソフトウェア開発のための手法である。アジャイル型によって、ユーザーの要求に柔軟に対応しながらソフトウェアを開発することが可能になったが、Web サービスや製品として開発されたソフトウェアでは、地域コミュニティで必要とされる細かな要求を満たせないという問題を抱えている。中にはソフトウェアをユーザーが細かくカスタマイズできるようにしているものもあるが、1 つの製品として開発する以上、限界がある。

一方、オープンソース・コミュニティでは、開発者が自ら必要とするソフトウェアが多く作られるという特徴がある。オープンソース・コミュニティでは、既存のソフトウェアでは実現できない細かな要求があった場合に、自ら開発することで対応している。しかし、前述したように、オープンソース・コミュニティは、ソースコードを理解できない人が参加することはできず、コミュニ

[*1] 類似の開発手法として、Scrum（1986）やエクストリーム・プログラミング（XP）[Bec99]（1996）などがある

ティにおいて開発者ではない市民と開発者が協働し、市民から寄せられる要求をもとに開発をするといったことはなかった。

2.5　共生プログラミング

これに対して、近年、NPO や市民による社会的なソフトウェア開発も行われるようになってきた。石田らは、筆者らも参加したパネルディスカッション [高崎 06] において、開発者と利用者が入り交じったコミュニティを形成し、そのコミュニティにおいて参加者自身のためのソフトウェアを開発する手法として「共生プログラミング」を提唱した。石田は、共生プログラミングと工業的なソフトウェア開発の違いを以下のように指摘している。

共生プログラミング

- 明確な発注者がいない：仕様はコミュニティの中でむにゃむにゃと決まる
- 明確な期限がない：いつまでも作り続ける
- 効率を考えない：例えば、キャンプファイヤーをやりながらソフトが開発される
- ソフト開発がコミュニティと一体化している：自分が使うソフトしか作らない

NOTA のコミュニティにおける開発もこの「共生プログラミング」が行われているといってよいだろう。しかし、共生プログラミングの概念はまだ明確ではなく、その定義も曖昧である。本論文では、NOTA の事例における開発スタイルを「共生開発スタイル」と名付け、共生プログラミングを含む概念として定義している。第 3 章において、NOTA の開発形態を事例として共生開発スタイルについて詳しく述べる。

2.6　まとめ

本章では、本研究に関連する研究をソフトウェアの開発形態の側面に着目して紹介した。従来の Web サービスや製品として開発されたソフトウェアでは、地域コミュニティで必要とされる細かな要求を満たせないという問題を指摘した。地域コミュニティにおいて市民が開発に参加することで、それらの問題を解決していく必要性があると考える。

第3章

共生開発スタイルの提案

本章では、市民やNPOによる社会的なソフトウェアの開発アプローチである「共生開発スタイル」を提案する。本研究を進める過程において、横浜市でNOTAを利用して共生開発スタイルが広がったプロセスを説明する。

3.1　はじめに

本章では、市民や NPO による社会的なソフトウェアの開発アプローチについて述べる。私が開発した NOTA は、地域のコミュニティ活動と持続的な関わりを持ちながら、開発者以外の人も参加する開発コミュニティにおいて、開発と普及が進められた。本研究では、このような開発者－普及者－利用者が一体となった共生的なコミュニティによる開発を、「共生開発スタイル」と呼び、その概念を詳しく述べる。

また、本研究において、2005 年 10 月から形成された NOTA の開発コミュニティである「NOTA ネットワーク」について述べる。NOTA ネットワークは、NOTA の開発を行うと同時に、メンバーらが横浜市の市民活動において積極的に用いたために、現在約 2000〜3000 人のユーザーが利用している。本研究の共生開発スタイルの提案は、筆者が実際に関わった横浜市における NOTA ネットワークの活動から得られた知見を元にしている。

3.2　共生開発スタイルの提案

3.2.1　共生開発スタイルの定義

本研究は、市民がソフトウェアを開発し、コミュニティで運用する中で得られるフィードバックを元に継続的に改良していく開発形態を「共生開発スタイル」と定義する。地域コミュニティにおいては、システムの選択と開発のプロセスは、従来のソフトウェア開発とは異なるものになる。コミュニティ活動支援システムを研究するにあたり、システムだけでなく、システムの背後にあるコミュニティ活動や開発手法も研究の対象としている。なぜなら、システムはコミュニティにおける人的なネットワークや社会的なコンテキストの中に位置づけられ、それらの理解なしには、有効なシステムは開発できないと考えるからである。

3.2.2　共生開発スタイルの特徴

以下に、石田が指摘した共生開発スタイルの各項目について詳しく述べる。参考のため、工業的な開発と共生開発スタイル比較を表 3.1 に示す。

共生開発スタイルの特徴

- 明確な発注者がいない：仕様はコミュニティの中でむにゃむにゃと決まる
- 明確な期限がない：いつまでも作り続ける
- 効率をあまり考えない：例えば、キャンプファイヤーをやりながらソフトが開発される
- ソフト開発がコミュニティと一体化している：自分が使うソフトしか作らない

明確な発注者がいない

工業的なソフトウェア開発においては、発注者は対価を払ってソフトウェアの開発を依頼するが、共生開発スタイルにおいては、開発者と利用者が 1 つのコミュニティを形成しており、開発の仕様も開発者だけが決めるわけではない。プロセスとしては、複数の組み合わせがあるが、アジャイル開発のように小さなプロジェクトを反復的に繰り返し、ソフトウェアを開発するという特徴が見られる。そのプロジェクトの中では、要求定義や開発、バグ修正など作業に明確な担当者は存在しない。NOTA ネットワークの場合は、運営方針として「この指止まれ方式」で進めることを決めている。この指止まれ方式とは「何かをやりたい人がメーリングリストや会合で提案し、賛同者が集まればその人を中心に行動するというやり方のこと」である（付録 B NOTA ネットワーク運営方針を参照）。

明確な期限がない

開発には、明確な期限が設けられていない。ただし、ワークショップやセミナーなど対外的なイベントが行われると、期限を設定することもある。プロジェクト全体で、完成の基準が曖昧であるため、期限が決められないともいえるだろう。発注者との金銭的な授受がないことや、ビジネス的な競合との競争がないことが原因として考えられる。ただし、完成の基準が明確でないとはいえ、完成度が高くなければ、実際には使用されない。そのため、開発に関わる者には、コミュニティ内での期待が寄せられるため、完成度をあげるための圧力は存在するといえるだろう。共生開発スタイルにおいては、コミュニティの活動が本流としてあり、ソフトウェアの開発は、それに準ずるものとして位置づけられる。

効率をあまり考えない

第 2 章で挙げたような工業的な開発手法は、いずれも、期限内にできるだけ効率的に開発できるようリスクを軽減しながら開発するために考案された。しかし、地域コミュニティにおける共生開発スタイルでは、効率は重要ではない。多少効率が悪くてもメンバーの参加意欲が重視される。参加意欲は、プロジェクトに対する自身の仕事の貢献度合いや、プロジェクトの構成員の魅力、プロジェクトを通じて得られる学習度合いなどによって高められる。

ソフト開発がコミュニティと一体化している

繰り返し述べたように、地域コミュニティとの持続的な関わりを持ちながら開発が行われることが大きな特徴である。開発は、コミュニティ活動の中に組み込まれており、活動と開発は相補的な関係になる。コミュニティのメンバーはある程度固定されているが、すべてのメンバーが定常的に作業をするわけではない。他人に依頼されたり、自分から名乗り出たりして、興味のあるプロジェクトに参加する。

表 3.1　コミュニティ活動支援システムの分類

利用目的	ビジネスの遂行支援	仮想的なコミュニティにおける情報交換	地域活動支援
対象	企業ユーザー	エンドユーザー	地域コミュニティ
開発手法	ウォーターフォール型など	アジャイル型など	共生開発スタイルなど
開発主体	企業	企業	地域コミュニティ
開発者と利用者の区分	明確	明確	曖昧
評価	個々人に対するアンケート等による認知心理学的評価	同左	コミュニティにおいて事例や組織を含めて総合的に評価
ユーザー特性	リテラシーがあることが前提	先進的なユーザーから使い始める	地域活動をする人が利用し、普及する

3.2.3　共生開発スタイルのメリット

共生開発スタイルを行うメリットは以下の通りである。ここでは、従来の開発スタイルと比較して優位点を述べる。

共生開発スタイルのメリット

- 効率をあまり考えずにシステムを継続して開発できる。
- 利用者のフィードバックがただちに得られる。
- 開発者とコミュニティが一体になっているため必要な機能だけを作ればよい。
- 高いモチベーションを保って楽しく開発できる。

効率をあまり考えずにシステムを継続して開発できる

すでに指摘したように、工業的な開発手法は、ビジネス的な要請からできるだけ速く少ない経費で効率的に開発する必要があるが、共生開発スタイルにおいてはその必要がない。そのため、効率をあまり考えずに開発することができ、プロジェクトは経済的な理由で終了することがない。

利用者のフィードバックがただちに得られる

ソフトウェア開発とコミュニティの活動が一体になっているため、開発したソフトウェアはすぐにコミュニティ内で試験的に利用され、フィードバックを得ることができる。オープンソースコミュニティにおいては、一部の先進的なユーザーからのフィードバックを得ることはできるが、一般のユーザーからのフィードバックを得ることは難しい。共生開発スタイルでは、ソフトウェ

ア開発以外のコミュニティ活動に参加しているユーザーもソフトウェアを利用するため、フィードバックを得ることができる。

開発者とコミュニティが一体になっているため必要な機能だけを作ればよい

工業的な開発においては、顧客のために開発することが当たり前だった。しかし、共生開発スタイルでは、コミュニティ内で要求があるものだけを開発するため、自分もしくは自分に近いユーザーのために開発することになる。

高いモチベーションを保って楽しく開発できる

共生開発スタイルにおける開発者のモチベーションは高い。コミュニティ内でユーザーの要求があるものだけを開発するため、ユーザーの顔が見えることや、そもそも興味のあるプロジェクトにしか参加しないことが、その要因である。

3.3　NOTA ネットワーク

3.3.1　NOTA ネットワークの形成

筆者は、NOTA という情報共有システムを開発する過程で、2005 年より横浜市都筑区において地域コミュニティに NOTA が持ち込まれ、有志により開発コミュニティが形成され、共生開発スタイルが広がっていく過程を経験した。

NOTA が横浜市のコミュニティに広がった過程は以下の 4 つの段階に分けることができる。1 つずつ説明する。より詳細なデータは「付録 A NOTA ネットワークの活動履歴」に記載した。

共生開発スタイルの経過

1. 開発者が一人で開発していた段階
2. 横浜市において一部のコミュニティに利用され始めた段階
3. NOTA の開発コミュニティ「NOTA ネットワーク」が形成され、共生開発スタイルが始まった段階
4. NOTA を拡張するプラグインがコミュニティ内で開発され多くのコミュニティに普及した段階

なお、本節における引用文は、共著『科学技術実践のフィールドワーク』の「学習環境のデザインのためのネットワーク指向アプローチ」[上野 06b] から抜粋したものである。

開発者が一人で開発していた段階（2003 年〜2004 年 12 月）

NOTA は、「Web のコミュニケーションを容易にする」というコンセプトのもとに私が一人で開発していた。

> 筆者の一人が NOTA を初めて知ったとき、ある IT ベンチャーが、この開発者と学生グループのために、NOTA の普及、展開、ビジネス化を担当していた。この IT ベンチャー

は、まず、ある商用サイトで100アカウントをフリーで配布した。しかし、ここでNOTAはほとんど使われることはなかった。例えば、このITベンチャーのNOTAの営業担当者によれば、このサイトのほとんどのユーザーは、一度だけ少し落書きや書き込みをするだけで、それ以降、NOTAにアクセスすることはなかった。こうしたことから、従来的なビジネスモデルの観点で、NOTAはこの企業内で収益の出ないシステムとみなされた。その中で、担当者は苦闘していた。

横浜市において一部のコミュニティに利用され始めた段階（2004年12月〜2005年冬）

NOTAは、横浜市において「市民グループの間で、さまざまなコミュニティがコラボレーションしたり、お互いの存在や活動を可視化するためのツールの要求」があり、そのためのツールとして利用され始めた。

そのようなとき、2004年12月、たまたまこの論文の執筆者の一人が、京都でのある研究会でNOTAの存在を知り、この京都の学生グループの一人からアカウントを提供してもらい、地域の市民グループに紹介した。まず、筆者は、地域のWeb技術に詳しく、技術を地域に普及するのに熱心だった市民にNOTAを紹介し、また、アカウントの提供のアレンジを行った。この市民は、NOTAについての情報をただちに地域のメーリング・リストで流し、これを見た他の何人かの市民もNOTAのアカウントを得た。筆者がNOTAを紹介して1ヶ月以内に少なくとも3つのNOTAサイトが作られ、そこへ1000から2000のアクセスがなされた。

メーリング・リストや直接のコミュニケーションを通して、たちまち、NOTAは地域ネットワークの中で有名になり、さまざまなNOTAサイトが作られるようになった。横浜の地域の市民グループは、いろいろなテーマでのコミュニティを形成しており、大学ともいろいろな活動を通して、もともとつながりがある程度できていた。このような状況の中で、市民グループの間では、さまざまなコミュニティがコラボレーションしたり、お互いの存在や活動を可視化するためのツールの要求が高かった。こういうことから、ここ2、3年、市民グループは、Blog、Xoops、Wikiなど自らサーバーにインストールして、試験的に運用、あるいは、実際の運用を行ってきた。こういうところにNOTAは紹介されたのであり、まさに、NOTAは、市民グループにとってさまざまなことを可能にし、約束する道具だったのである。

NOTAの開発コミュニティ「NOTAネットワーク」が形成され、共生開発スタイルが始まった段階（2005年冬〜2006年春）

NOTAの利用方法や改良方法に興味を持つ有志が集まって地域コミュニティの活動とリンクした開発コミュニティを形成し、共生開発スタイルが始まった。

実際にこれまで見てきたような形でNOTAがさまざまに用いられる中で明確にオープンソース的なコミュニティを形成していこうという方向が出てきた。2005年の初夏に、

NOTA 開発者の永田および関係していた IT ベンチャーが、上野にコンタクトをとり、NOTA の技術的な情報を提供し、また、NOTA の使用例を紹介する NOTA ポータルサイトを構築することを提案した。こうしたことをきっかけに、地域市民グループ、永田、IT ベンチャー、学生、院生、上野などが参加して、数回の会議が行われ、「NOTA ネットワーク」と呼ばれるコミュニティが形成された。この会議の中で、NOTA のプログラムを「NOTA ネットワーク」の参加者にフリーで配布すること、ポータルサイトは、この会議のメンバーによって運営していくことが決定された。この会議、および、メーリングリストの中で、NOTA 使用上の規約なども議論された。

NOTA を拡張するプラグインがコミュニティ内で開発され多くのコミュニティに普及した段階（2006 年春〜2007 年）

NOTA は、汎用性の高いシステムだったため、NOTA ネットワーク内で無料で配布して以後、さまざまな形で工夫して用いられていたが、NOTA 自体を拡張できるようにするため、2005 年 11 月にプラグイン・プロジェクト・ミーティングを開き、NOTA をどのように拡張していきたいかが話し合われた。私は、2006 年夏より共生開発スタイルによる開発を促進するために、利用者が NOTA を拡張できるプラグイン機能を追加した。この結果、NOTA を応用した地図システムや、ビデオ会議システム、障害者コミュニケーションシステムなどが地域コミュニティ内で開発された（詳細は第 6 章を参照）。2006 年 9 月には横浜市市民活動支援センターにおける IT 講習会で NOTA の講座が行われた（図 3.1）。

図 3.1　横浜市市民活動支援センターにおける NOTA の講座の様子

3.3.2　NOTA ネットワークの実践

上述したように、NOTA の開発コミュニティ「NOTA ネットワーク」には、さまざまな立場の人が参加している。もともと、市民のコミュニティから派生したコミュニティであるため、当初から「プログラマーではない人も参加できるオープンソース的ネットワーク」であることを表明していた。現在のメンバーは、横浜の市民、大学の研究者、学生、小学校教諭など約 50 人で構成されている。これは、Linux などのいわばプロ集団に閉じたオープンソースではなく、新しい形態のオープンな開発コミュニティを志向している。

NOTA では、開発が容易な Flash アプリケーションやクリップアートを NOTA 上で簡単に貼り付けて動作させることができるため、今まで開発に参加したことがない人たちが開発に加わることができる。NOTA が Web アプリケーションの基盤を提供し、ユーザーが求めるアプリケーションを普及させるための触媒として働く。その結果、ただちにコミュニティ活動で利用可能なソフトウェアが開発できる。

また、NOTA ネットワークおよび周辺のコミュニティでは、市民活動の支援のためにワークショップや教室、地域イベントなどで NOTA の利用について他の市民に伝えている。例えば、2006 年 3 月には横浜市の「つづき人フェスタ」で NOTA の展示をした（図 3.2）。この展示のために NOTA ネットワークのメンバーが協力した。ノートパソコンを持ち込んでその場でワークショップをしたり、大型タッチパネルディスプレイを研究室から運んで NOTA の体験ブースを設けたりした。会場には、市民活動をしている市民や、お年寄り、子供が立ち寄った。このとき、NOTA ネットワークでは、ユーザー、サポーター、エンジニアと書かれた 3 種類のバッジを制作し、配布した（図 3.3）。バッジの数は、エンジニアが最も少なく、サポーターとユーザーがほぼ同じである（比率は、ユーザー：サポーター：エンジニア＝ 5:4:1）。実際には、デザイナー、プログラマー、サポーター、サーバー提供者、市民活動のリーダー、利用者、研究者などに細かく分類できるが、NOTA ネットワークのおおよその構成を示しているといえる。

3.4　まとめ

本章では、市民や NPO による社会的なソフトウェアの開発アプローチである「共生開発スタイル」と、横浜市における NOTA の開発コミュニティ「NOTA ネットワーク」の形成と実践について述べた。NOTA は、さまざまな用途に利用できるプラットフォーム志向の設計を持っていたコミュニティに受け入れられ、またコミュニティの活動によって NOTA 自体もそれに合わせて改良が加えられた。共生開発スタイルに関するより詳しい考察は第 8 章で行う。

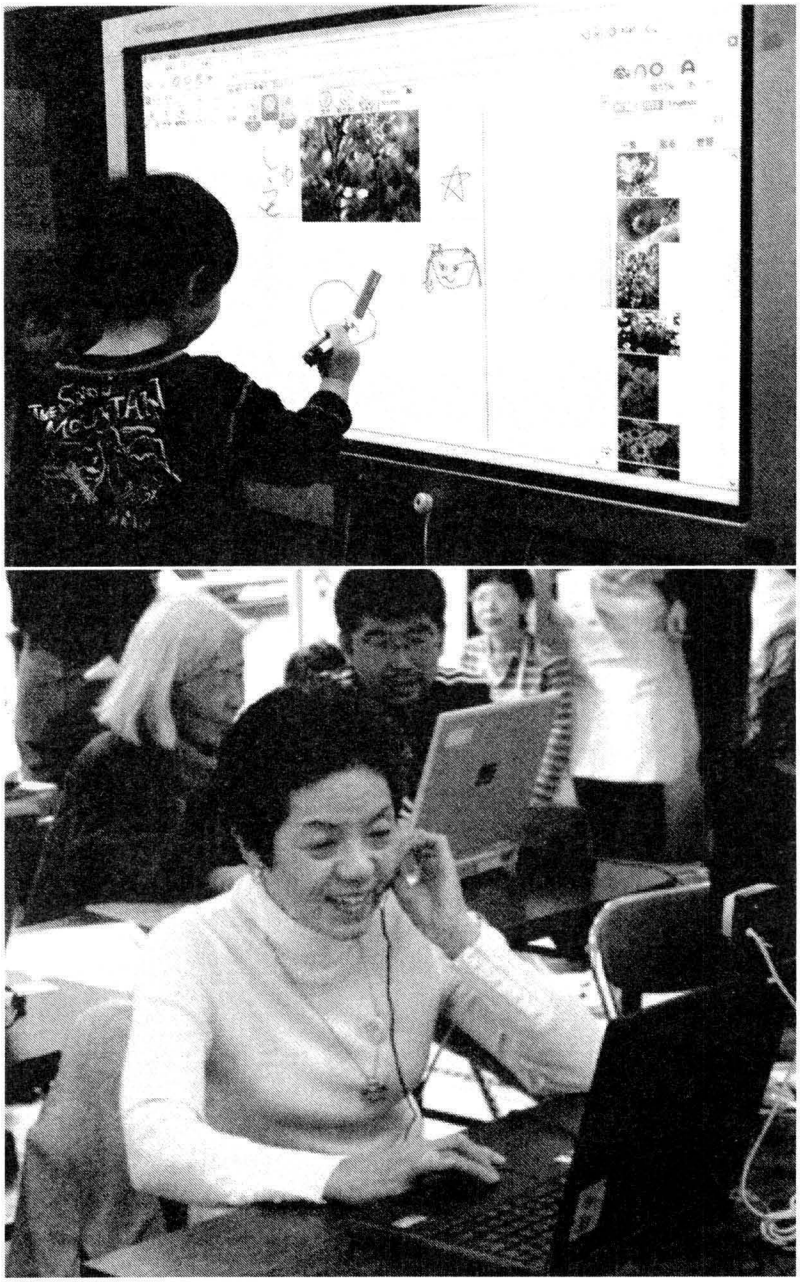

図 3.2　つづき人交流フェスタの活動の様子

明日を語るなら、自由なキャンバスで。
サポーター
明日を語るなら、自由なキャンバスで。
サポーター
明日を語るなら、自由なキャンバスで。
NOTA
エンジニア

図 3.3　NOTA バッジ

第 4 章

NOTA の設計と開発

本章では本研究において開発した NOTA のシステムについて述べる。NOTA の使い方を示し、システムの新規性を以下の 4 つに分類し、説明する。NOTA をユーザーが拡張することができるアプリケーション・プラットフォームの機能については、特に重要な部分なので、第 5 章で詳しく述べる。

- リアルタイム編集の実装
- ユーザーアカウントと編集権限
- HTML との互換性とデータ構造
- 手書き線の実装

4.1　はじめに

近年、ブラウザ上でウェブページを編集したいという要求から、ブラウザから Web に画像をアップロードして画像を共有できる掲示板システムや、ブラウザ上でページの内容の編集を行うことができるシステムが普及したりするなど、動的なウェブページの作成システムが普及しつつある。ブラウザ上で動的にウェブページを作成することは、従来のようにクライアントで作成したページをサーバーにアップロードする手法と比較して、以下のようなメリットがある。

- ウェブブラウザがあればどの端末からでも利用できる。
- クライアントのソフトウェアの設定が不要である。
- 複数人でページを協調的に編集できる。

しかし、今日普及しているこれらのシステムでは、編集のインタフェースとして CGI と HTML フォームを連携させて用いることが多く、クライアント側で用いる一般のホームページ作成ソフトウェアと比較して、インタフェースと機能に多くの制約がある。このため、直接操作を用いて WYSIWYG な編集環境を実現したり、リアルタイムでページの更新を行ったりするシステムを構築することは難しかった。

これに代えて、ブラウザ上で手軽に編集できるというメリットを活かしながら、上記の欠点を克服したソフトウェアを開発することが求められている。本論文では、Wiki と同様の手軽さで、リアルタイムに動的なページの更新を行い、かつ自由なレイアウトのページを直接操作によって作成できるウェブページの共有システム「NOTA」を提案する。本システムは Flash をサポートする一般のブラウザ上で動作する。HTML の知識がなくとも、テキスト、画像、図形が混在したページを、ドラッグ＆ドロップを多用した分かりやすい操作方法により、複数のメンバーで協調的に作成および管理することができる。NOTA におけるページの編集のワークフローを図 4.1 に示す。

4.2　従来のシステム

ローカル HTML ページを作成し、サーバーにアップロードする従来の方式には、二つの欠点がある。ローカルで管理するため、ページを修正するたびにアップロード作業が必要であることと、ローカルページが共有できないため、複数人での編集作業が難しいことである。このため、Web 上で手軽にページを更新したり、複数人で協調的にページを作っていくことは困難だった。これらの問題を解決するために、ブラウザ上で CGI、Java 等を利用し、ページの編集を可能にするシステムが考案された。その代表例として Wiki がある。

Wiki は、ブラウザ上で HTML ページ全体の編集が行える点と、HTML タグを使用せずに簡単な記号を記述することで、段落の修飾や、動的なリンクの作成が可能である点が優れている。ページの編集は各ページの最上部にある [編集] というリンクを開き、HTML の FORM タグを利用したテキストボックスを表示させ、その中で本文を編集し、[送信] もしくは [保存] ボタンを押すことによって行う（図 4.2)。Wiki には、編集開始と編集完了後の送信の間の時間差があるため、同

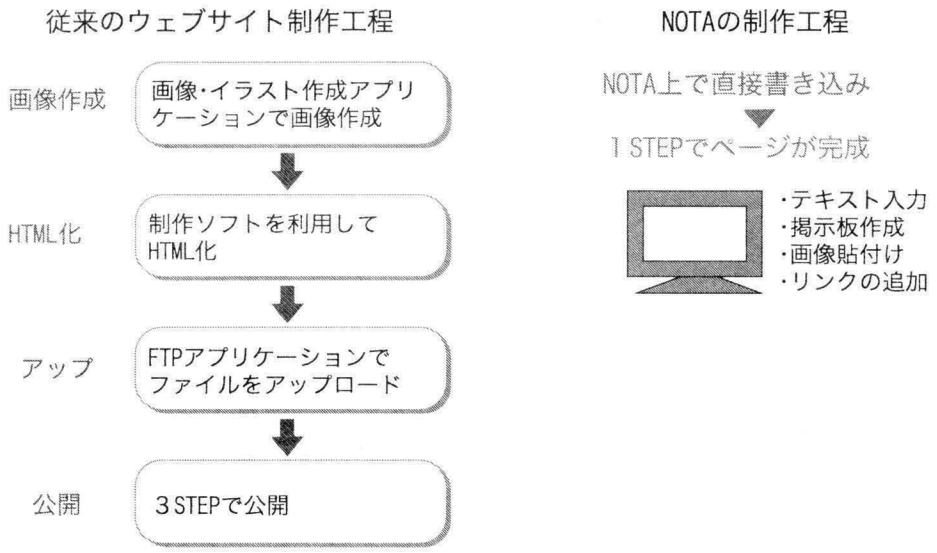

図 4.1　NOTA の編集のワークフロー

時に複数人が編集作業を行った場合には編集内容の衝突が起こる。この問題の回避策はない。また、Wiki は本文の編集は簡単に行えるが、文の修飾を変更したり、画像を貼り付けたりすることは難しい。多くの Wiki プログラムでは、定められたタグで文字を囲んだり、別途ファイルをアップロードし、ファイル名を指定することでこれらを実現しているが、テキストベースであるため、GUI に親しんだユーザーには受け入れられず、結果として Wiki はこれらのユーザーにはあまり利用されていないといえる。

4.3　NOTA

前述のような問題を解決するために、NOTA というシステムを開発した。NOTA は、従来の HTML を利用したウェブページの動的作成システムの問題を解決し、Flash を用いた直感的なインタフェースを持つ Web の共有システムである。NOTA を利用することにより、Wiki 同様の手軽さで、表現力が高いページの作成・共有を手軽に行える。すなわち、マウス操作による WYSIWYG のページの編集機能を持ち、手書き線や、図形の貼り込みなどが簡単に行える。また、送信コマンドを押さなくても非同期でサーバーと通信し、編集するたびに保存および更新を行う。これによって、複数のクライアントで編集中の画面を共有することができ、複数のメンバーで協調的に作成および管理することができる（図 4.3）。

Web における動的なページの作成と共有を可能にするためのソフトウェアとして Flash を用いた。Flash が非常に多くのプラットフォームをサポートし、ウェブページの不可欠な部分を占めていることに着目し、Flash と CGI を連携させ、動的なウェブページの作成を行うことを試みた。具体的には以下の機能を持つ。

- すべての書き込みが Web 上で WYSIWYG に共有可能である。

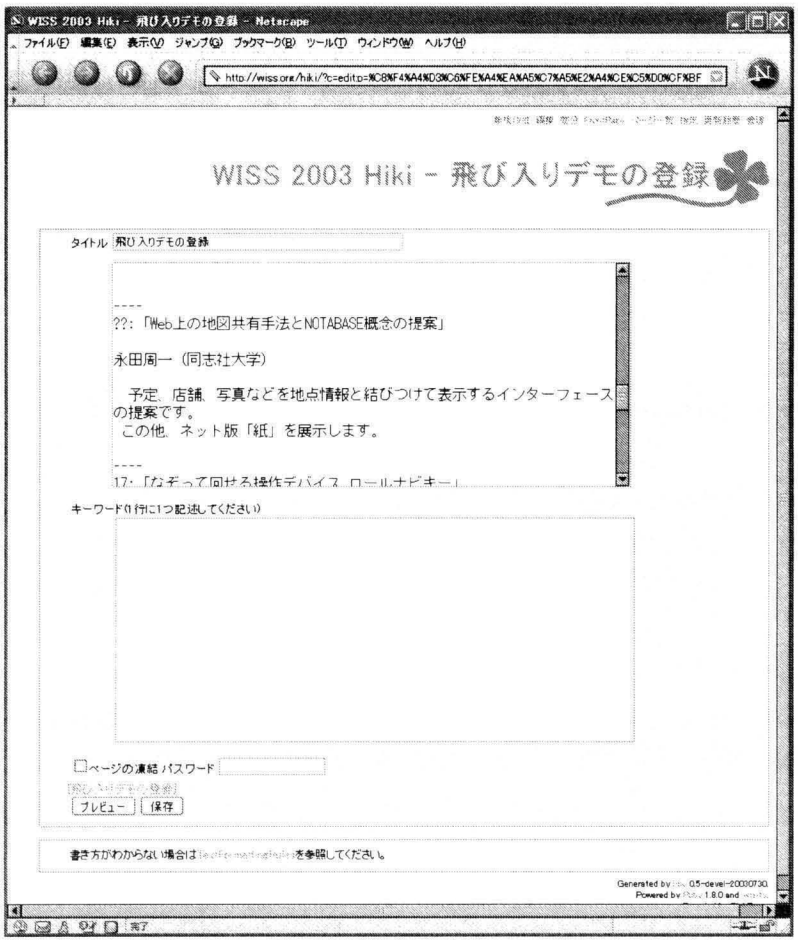

図 4.2　一般的な Wiki の編集画面

- 画面上のどの位置に対しても文字を貼り込むことができる。
- ドラッグにより手書きで線を引くことができる。
- 基礎的な図形作成機能を備える。
- JPEG, PNG, GIF, BMP などの画像をアップロードし、どの位置にも自由に貼ることができる。Flash は、外部の画像の読み込みで JPG のみをサポートしているので、画像をアップロードした際に、ImageMagick を用いて、画像の形式を JPG に変換している。
- Windows、Mac、Linux、携帯端末（一部）などほぼすべてのプラットフォームでそのまま動作する。
- ページ間にリンク構造を持たせ、容易にページ間でリンクを作成できる。リンクを構築する際に、Wiki と異なり、単語によるリンクは採用しなかった。NOTA でリンクを張るには、既存のページの一覧からページを選ぶだけでよく、自動でリンクを貼る機能が必要でないと考えたからである。
- 作成者の情報を記録し、オブジェクトを作成者単位で管理できる。

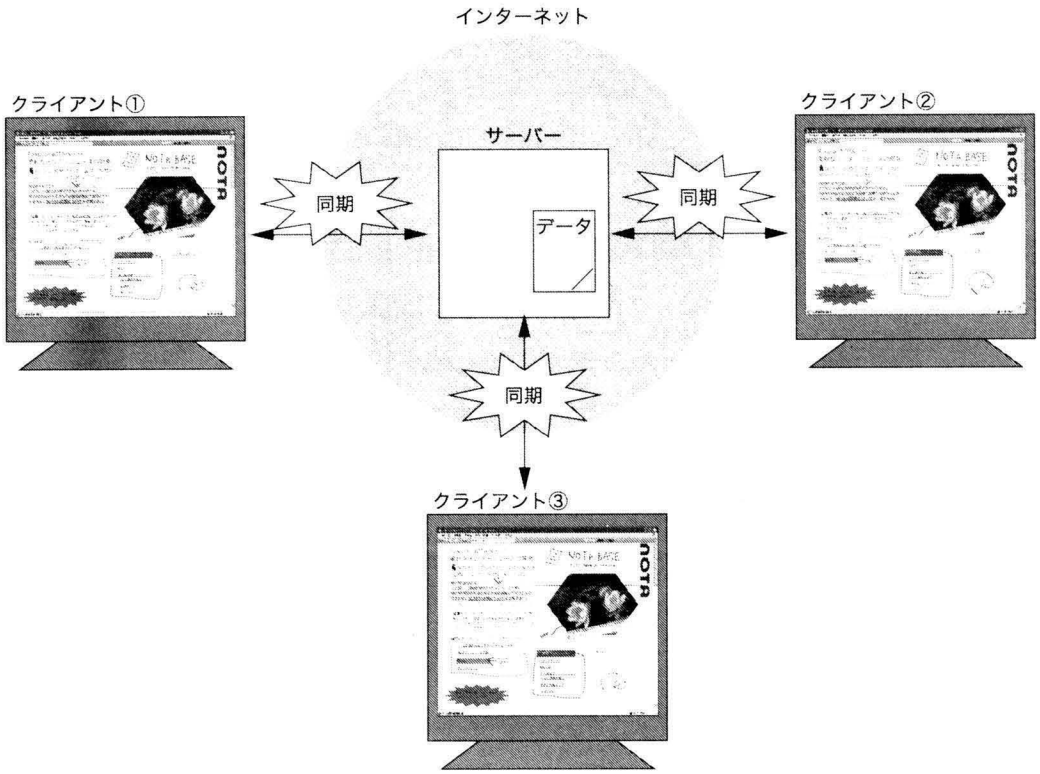

図 4.3　NOTA の協調的編集の概念図

- 複数人で同時に書き込みを行った場合、他人の編集がただちに反映される。
- PDF 並の印刷機能を持つ。

4.4　NOTA の使い方

NOTA の編集画面を図 4.4 に示す。画面上部の「ツールバー」、画面右の「一覧」、そして画面中央の「ページ本文」の領域から構成される。NOTA の使い方を説明する。

4.4.1　基本ツール

以下に編集画面の各領域について、簡単に述べる（詳細は各機能の解説で述べる）。

1. 「編集」ツール
   画像などのオブジェクトを選択し、移動・編集するために使用する。
   編集画面に切り替えた直後は、このツールを選択した状態になる。
2. 「ペン」ツール
   手書き線の入力を行うために使用する。
   このツールが選択されていると、ツールバー下部に「色」「太さ」が表示される。

図 4.4　NOTA の編集画面の例

3. 「消しゴム」ツール
   手書き線を消すのに使用する。
4. 「図形」ツール
   丸・三角・四角・星型など、多彩な図形を描画する際に使用する。
   このツールが選択されていると、ツールバー下部に「色」と図形の選択メニューが表示される。
5. 「文字」ツール
   文字を入力する際に使用する。
   文字の色、サイズ、フォントをワンクリックで簡単に変更できる。文字を選択していない状態で色を選ぶと、テキストボックス全体の背景色となる。なお、NOTA はユニコードに対応しているため、多言語の入力もできる。
6. 「一覧」
   ページやアカウントの管理、画像・ファイル・プラグインの貼り付け、全ページを対象にした検索などを行う。

7.「ページ本文」
編集領域である。ここに文字を入力したり、手書き線を入力したりする。

4.4.2　認証

「閲覧画面」から「編集画面」に切り替えるには、ツールバーの「編集開始」ボタンを押す。パスワードの入力ダイアログが表示されるので、サイトの管理者から発行されたユーザー名とパスワードを入力する。

4.4.3　写真の貼り付け

次に、NOTA のページ上に画像を貼り付ける方法を述べる。ローカルの画像を選択し、アップロードすると、ただちにページ中央に画像が貼り付けられる。画像のサイズや位置は自由に決めることができる。

1. 画面右の「貼る」タブを選択して、「参照」ボタンを押す（図 4.5）。
2. 「ファイルの選択」ダイアログが表示されるので、ページに貼り付けたい画像ファイルを選択する。
3. 「参照」ボタンの左側に貼り付けたファイルのパスが表示されるので、確認後「貼り付け」ボタンを押す。

NOTA に貼り付けられた画像は自動的に JPEG 形式で圧縮され、幅 700 ピクセル、50KB 前後のサイズになる。しかしディスプレイ上では元画像との差はほとんど分からない。美しく見せたいときには、「高精細」というチェックボックスを有効にしてから「貼り付け」ボタンを押す。高精細で貼り付けられた画像は、幅 1000 ピクセル、200KB 前後のサイズに圧縮される。デジタルカメラで撮影した風景写真などを貼り付けるときには「高精細」を有効にした方がよい。

対応画像ファイルは、JPEG、PNG、TIFF、BMP、PICT 形式のファイルである。アニメーション GIF、透過 GIF、透過 PNG 形式には対応していない。

貼り付けた画像の大きさを変更するには、図形や文字領域の大きさ変更と同様、右下の「サイズ調節つまみ」をマウスでドラッグして行う。元画像の縦横比を維持してのサイズ変更を行いたいときには、Shift キーを押しながらドラッグするか、画像の左上と右下の対角線上にマウスをドラッグする。縦横比を維持した位置のときにはマウスカーソルが吸着したような動作をする。

4.4.4　ファイルの貼り付け

NOTA では、貼り付けるファイルとして画像ファイル（JPEG、PNG、TIFF、GIF 形式など）が選択された場合にはページ上にそのままイメージとして貼り付けられる。画像以外のファイル（PDF、DOC、XLS 形式など）が選択された場合には、ファイル名つけのアイコンとして貼り付けられる。

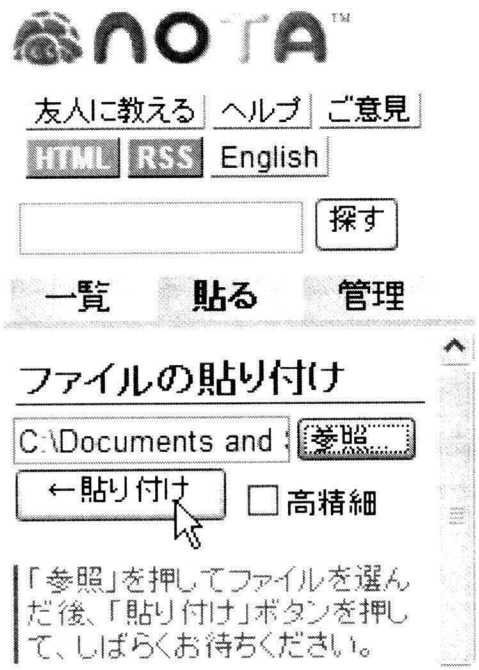

図 4.5　ファイルの貼り付け

貼り付けたファイルの種類によって、表示されるアイコンが異なる。ファイルの種類とアイコンの対応は以下となる。

4.4.5　手書き線の描画

NOTA の特徴の 1 つに、手書き線の描画機能がある。描画のインタフェースは一般的なものであるが、NOTA の場合は、Web 上でなめらかな線を描画するために独自のアルゴリズムを実装した。アルゴリズムについては、後述する。

画用紙にペンで絵を描くように、ページ内の好きな位置に、好きな色・太さで手書き線を書くことができる。ペンの先端部分が入力位置になる。ドラッグすることで、線が描画される（図 4.6）。他の色や太さに変更したい場合には、その都度ツールバーの「色」「太さ」から選択する。ペンタブレットを利用すれば、より精緻な絵を描くこともできる。以下に利用方法を示す。

1. ツールバーから「ペンツール」を選択する。マウスカーソルがペンに変更される。
2. ツールバー上部の「色」から好きな色を選択し、「太さ」からペンの太さを選択する。
3. 好きな位置にマウスやペンタブレットで入力する。

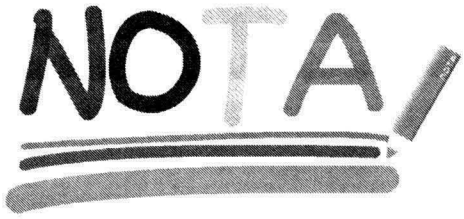

図 4.6　手書き線の描画

図 4.7　手書き線の消去

4.4.6　手書き線の消去

入力した手書き線を消去したいときには「消しゴムツール」を使用する。なお、「消しゴムツール」を選択している状態のときに画像や図形の上でドラッグ操作をすると、画面上ではそれらも消去されて表示されるが、実際には消去されない。「消しゴムツール」で消去できるのは「ペンツール」による手書き線だけである。 削除の方法を以下に示す。

1. ツールバーから「消しゴムツール」を選択する。マウスカーソルが「消しゴム」に変更される。
2. 消去したい手書き線の上をマウスでドラッグする（図 4.7）。

4.4.7　リンク

NOTA において「リンク」機能は重要な意味を持っている。Web が普及したのは、異なるリソース間を結びつけるリンク機能があったためとも考えられるが、これまでの HTML でのリンク作成はタグによるマークアップが必要で、HTML についての知識がない全くのパソコン初心者が行うにはいささか敷居が高く、操作も直感的ではなかった。

そこで、NOTA はリンクの作成・修正を誰もが簡単、かつ直感的に行えるような機能を実装した。NOTA でのリンクの設定方法は以下のように行う。

1. リンクを設定したいテキストをマウスで範囲選択する。

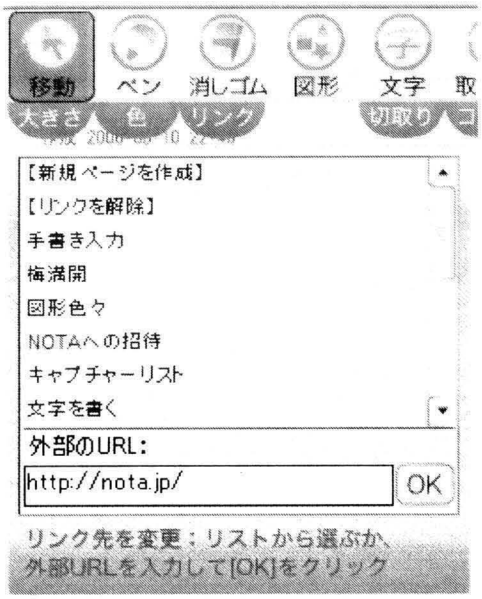

図 4.8　リンクの選択

2. ツールバー下部の「リンク」にマウスカーソルをあわせると、自動的にリンク設定メニューが表示される（図 4.8）。
3. NOTA サイト内のページへのリンクを設定したい場合には、リストからページを選択する。外部のページへリンクを張りたい場合には、「外部の URL」欄にアドレスを入力する。

なお、文字を範囲選択していない状態で、「リンク」からページを選ぶと、カーソル位置にページタイトルが挿入され、そのページに対するリンクを張ることができる。

4.4.8　リンクからの新規ページ作成

NOTA において、新規ページを作成する方法はツールバーの「新規作成」ボタンを押す以外に、もう 1 つ、リンクから作成する方法がある。

任意の文字を選択し、「新しいページへリンクを張る」を選ぶと、選択したテキストをタイトルとする新規ページが作成される。新規ページには、元のページへのリンクが自動で作成される。リンク先とリンク元が相互にリンクされることで、リンク構造が構築される。新規作成ボタンを押して作成した新規ページでは、ページのタイトルは常に「新規ページ（作成日）」となるが、リンク機能から作成したページでは、そのページへのリンクを設定したテキストの内容が自動的にページタイトルに設定される。

もうひとつの特徴として、リンク機能から作成した新規ページでは自動的に作成元ページへのリンクが「リンク元へ戻る」として作成される。つまり、設定元ページから設定先ページへの一方向リンクの作成だけでなく、自動的に相互リンクが作成される（図 4.9）。

リンクからの新規ページ作成方法は以下のように行う。

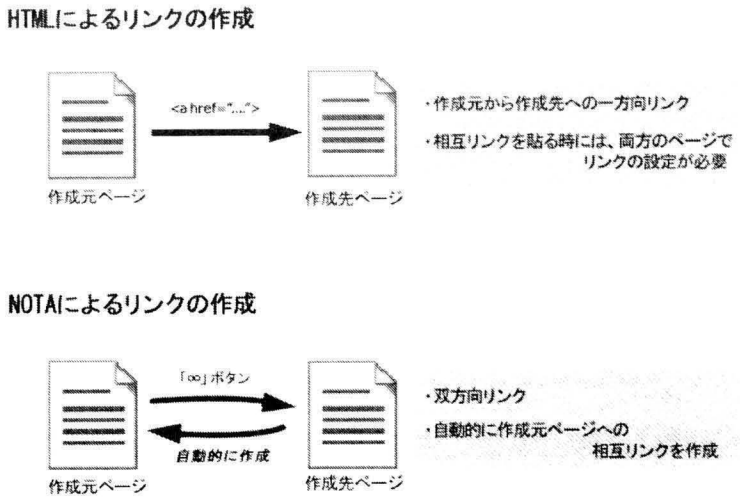

図 4.9　NOTA による相互リンク

1. リンクを設定したいテキストをマウスで範囲選択する。
2. ツールバー下部の「リンク」にマウスカーソルをあわせると、自動的にリンク設定メニューが表示されるので、「【新規ページを作成】」を選択する。

範囲選択したテキストをタイトルに持つ新規ページが作成され、「リンク元へ戻る」には自動的に作成元ページへのリンクが設定される。

4.4.9　オブジェクトの操作と部品の重なり順

部品の移動

部品を移動するには、部品選択時に左上に表示されている「移動」部分をマウスでドラッグする(画像・図形・プラグインでは、各部品本体のドラッグ操作でも移動可能)。削除するには、右上に表示されている赤い「×」印をマウスでクリックする。

部品の削除

誤って部品を削除してしまった場合には、ツールバーの「取り消し」ボタンを押す。「取り消し」ボタンを押すたびに、実行した操作が 1 つずつ取り消される（ただし、他のページに移動した場合には、移動前のページの操作を取り消すことはできない)。取り消した操作を復帰したい場合には、同じくツールバーの「取り消しを戻す」ボタンを押す。

色の指定と部品の重なり順

色は「文字入力領域」「図形」においてのみ設定が可能である。図形では塗りつぶし色を、文字入力領域では文字色と背景色を設定できる。また、「図形」と「画像」「挿し絵」には「透明効果」

をつけることもできる。

透明効果が有効にされた部品は薄い色で表示され、その部品の下に重ねられている部品を透過表示するようになる。設定方法は、どちらもツールバーの「色」「透」から行う。

異なる部品を何層にも上下に重ねることもできるが、より面積の小さい部品が自動的に上に表示される。 これにより、部品の重なり順を変更するコマンドを用いなくても、小さな部品が下に隠れることがないため、部品の操作がしやすい。

4.5　NOTA の実装

本節では、NOTA のシステムの新規性を 4 つに分類して、詳述する。

4.5.1　リアルタイム編集の実装

NOTA には、クライアント・ソフトウェアや CGI で用いられる「保存」や「送信」といったコマンドが、ユーザーに対して用意されていない。入力した文字や、手書きの線などは書き込みが行われた直後に CGI を通じてサーバーに送信され、保存される。同時に、ページを閲覧している他のユーザーの表示に更新が反映される。一般に HTML から CGI を用いるためには、HTML の FORM 要素内において Submit ボタンを配置し、それをマウスでクリックするか、キーボードのキーを押す必要があり、自動的にユーザーの入力を検知し、サーバーに送信することはできない。また、自動で送信され、サーバーに保存されたデータの更新を別のクライアントがユーザーからの入力を受けることなく行うことはできない。

サーバーに送信する際に、ユーザーによる保存や送信等の入力が必要なシステムは、いずれも、最新のデータを表示し、それを編集し、サーバーに送信して更新するという過程においてタイムラグが発生するので、同一データを異なる端末において同時に編集作業を行った場合、いずれかの更新は破棄されることとなる。

Web 上で動的なページの作成を行う際に、これらのコマンドを選択しなければ情報が送信（もしくは保存）されないのは、即時性や手軽さを考慮した場合に障害となると考える。NOTA では、文字の変更、図形の移動／サイズ変更／回転、画像の貼り付けなど内容が変更された直後にクライアントからサーバーに差分データを送信し、サーバー上に保存する。図 4.10 に、流れを示す。各クライアントは、サーバーのデータが更新されると、更新情報をサーバーから取得し、クライアントの描画を更新する。

NOTA においては、部品ごとに更新日時データを持っているため、各クライアントが同時に別々の部品を操作しても衝突が起こることはない。また、同一の部品を操作する場合においても、あるクライアントが部品を移動し（X、Y 値を変更する）、同時に別のクライアントが部品を拡大・縮小する（WIDTH、HEIGHT を変更する）などのように異なる属性値を変更する場合は衝突は起こらない。同一の部品の同一の属性を二つ以上のクライアントが同時に操作した場合は、衝突が生じる。この場合は、時間的に前に操作したクライアントの操作は、後に操作したクライアントの操作によってデータが上書きされ、無効になる。

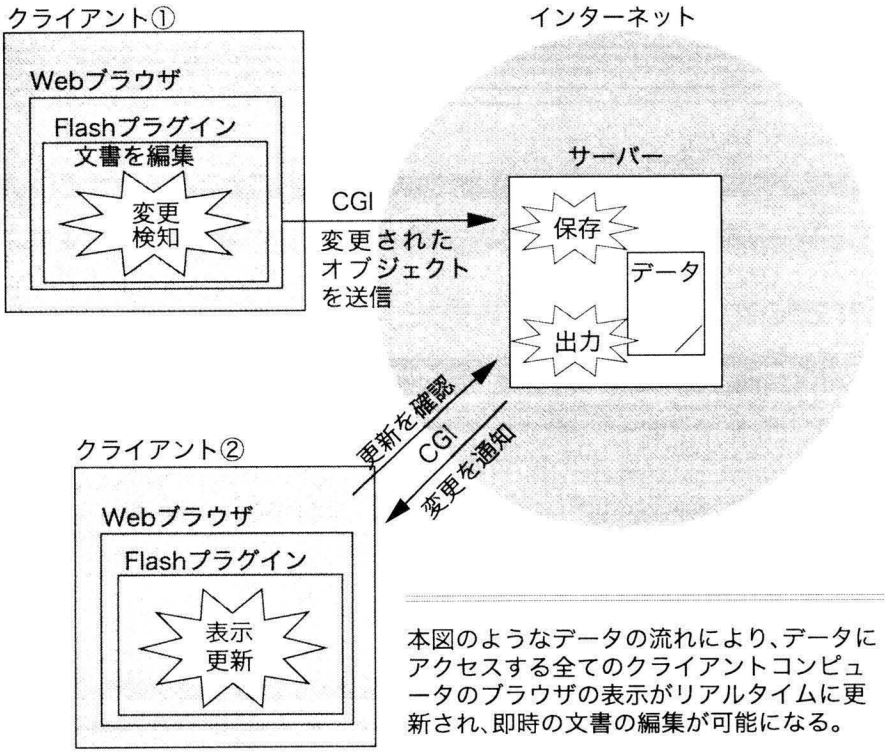

図 4.10　リアルタイム編集の流れ

4.5.2　ユーザーアカウントと編集権限

NOTA ではページ内の各部品について、誰が最初に作成したのかを簡単に確認することができる。確認するには、「編集画面」において、作成者情報を確認したい部品をマウスで選択すると、部品のすぐ上に作成者情報と作成日が表示される。更新された部品には、「作成日」の下に「更新日」が追加され、更新日時が表示される。

NOTA ではサイトを開設した管理人以外の人もページを編集することができる。しかし、ユーザーが操作可能な権限は 3 種類に分類されている。ユーザーアカウントを持っている人は、そのアカウントに許容された範囲内の編集操作を行うことができる。

NOTA に用意されているユーザーアカウントは「管理者」「会員」「ゲスト」の 3 種類に分かれている。それぞれのアカウントで実行可能な操作を表 4.1 に示す。

「管理者」権限のユーザーは、新規アカウントの作成や他人が作成したページの削除など、NOTA サイトの運営に関わる操作権限を有している。そのため、「管理者」権限は信頼できる人にのみ発行する。

表 4.1　アカウント別実行可能操作表

	管理者	会員	ゲスト
追加書き込み	可能	可能	可能
自分の部品の編集・削除	可能	可能	可能
他人の部品の編集	可能	可能	不可
新規ページの作成	可能	可能	不可
自分のページの削除	可能	可能	不可
他人の部品の削除	可能	不可	不可
他人のページの削除	可能	不可	不可
ページの凍結	可能	不可	不可
ユーザー管理	可能	不可	不可
第三者の閲覧制限	可能	不可	不可

4.5.3　HTML との互換性とデータ構造

NOTA は、既存の HTML による WWW との互換性を最大限保つように設計されている。NOTA のページ 1 つに URL が 1 つ与えられるので、HTML ページから NOTA のページに対し、リンクを張ることができる。逆に NOTA から外部の URL に対しリンクを張ることも可能である (図 4.11)。

データ構造として、サーバー内にページごとに XML ファイルでオブジェクトを管理している。ページ内で最上位に位置するテキストの一行目をタイトルとして利用するが、ファイル名は他と重複しない固有の ID で管理されており、タイトルが変更されたり、リンク元の文字列が変更されても、リンクが切れることはない。オブジェクトも固有の ID を持っており、ID ごとに作成者の情報を管理しているので、オブジェクト単位で作成者を識別できる。

4.5.4　手書き線の実装

NOTA は、Adobe Flash を利用することにより、マウスのドラッグ操作で Web 上で手書き線を描画できることが特徴の 1 つであった。しかし、マウスによる描画は、カーソル位置の検出点を直線で結ぶという仕様であったため、マウスの動作のぶれによって描画する線にもぶれが生じてしまうという欠点を持っていた。特に、ラップトップ型の PC において、タッチパネルを用いてきれいな線を描画するのは難しかった。これを解消するため、検出点のぶれを補正し、各点の間をベジェ曲線を用いて曲線でつなぐことで、なめらかで自然な手書き線を描画する手法を開発した (図 4.12)。これを用いて NOTA で実際に絵を描いた例を図 4.13 に示す。

マウスで描いた線をスムーズに補正するためには、多少のぶれは吸収するが、ユーザーの意図的なぶれ (方向転換や、角など) は確実に反映する必要がある。また、スムーズな線を生成するタイ

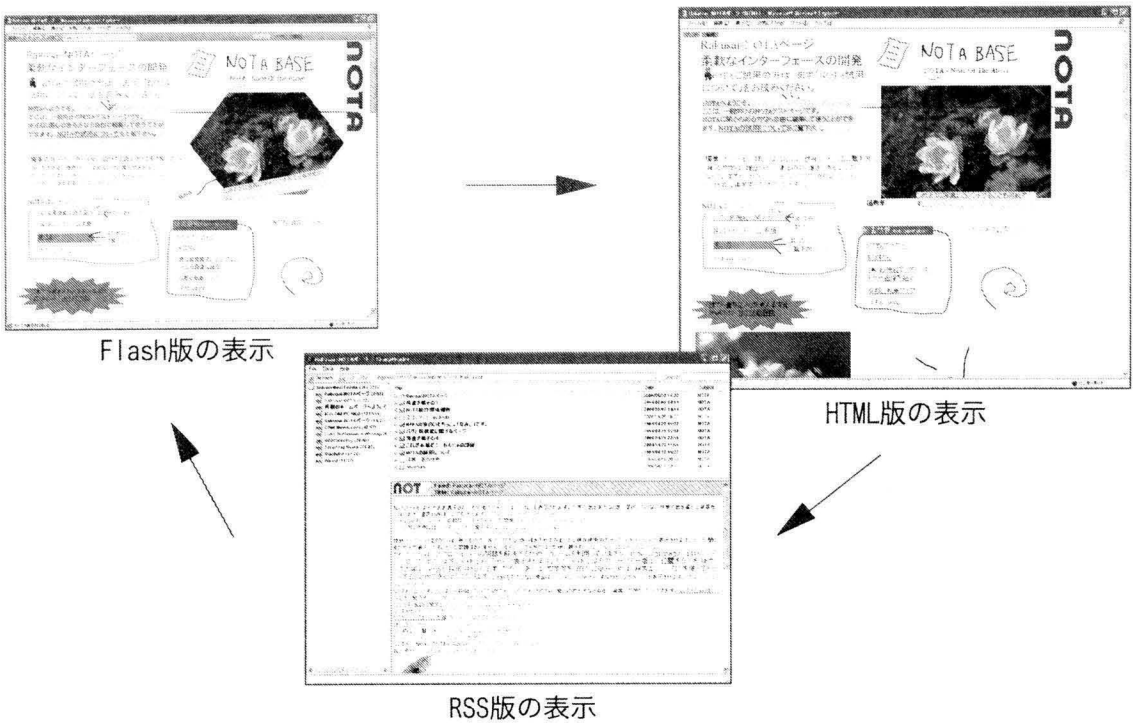

図 4.11　表示互換性の確保

ミングには、ユーザーがマウスのドラッグを終了したのち検出点をまとめて計算し生成する方法と、ドラッグしている途中で現在の検出点を元に、随時スムーズな線を生成していく方法がある。今回は後者を選択した。具体的なベジェ曲線のアルゴリズムについて述べる。まず、最初の検出点 A を記録する。次に検出点が変更されるたびに、現在の検出点 B と検出点 A を線分 C で結び、検出点 A〜B 間のすべての検出点について、線分 C との距離を測る。この距離が定義されたぶれ幅 D を超えたときに、その点 E を意図的なぶれとみなす。このとき、A-E 間で、線分 A-E とその間にある検出点の距離をすべて測り、最も遠い点 F を頂点とする曲線 A-F-E を描画する。これを繰り返し、なめらかな曲線を描画する。さらに、曲線と曲線のベジェの補助線の角度が水平に近いときは、補助線を水平に補正することで、曲線通しのつなぎをスムーズにしている。また、描画する線の太さにより、ぶれ幅 D の値を変化させることで、太さに応じた最適ななめらかな線のぶれの閾値を設定している。

スムーズな手書き線の描画は、Adobe Illustrator などで採用されているが、Flash 上で実現しているものはない。本手法はアルゴリズムも単純なため、Flash などの軽快さを求められる環境において利用可能である。また、同様の研究の例として、田川氏の「Afterglow」[田川] がある。

4.5.5　機能の比較

表 4.2 に NOTA と他のシステムとの比較する。

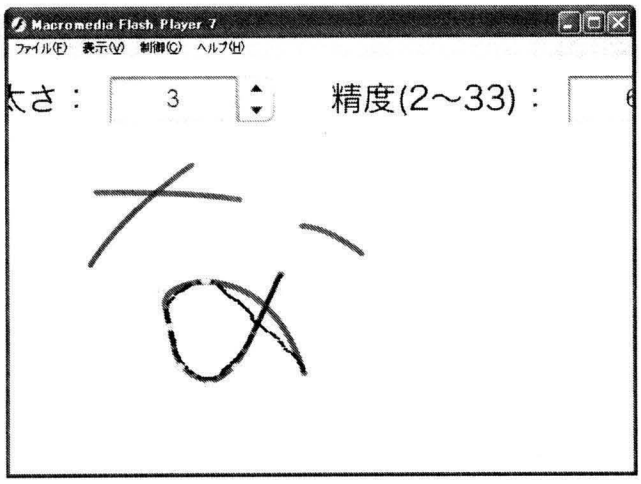

図 4.12　Flash 上で、スムーズな手書き線を描画するプロトタイプ

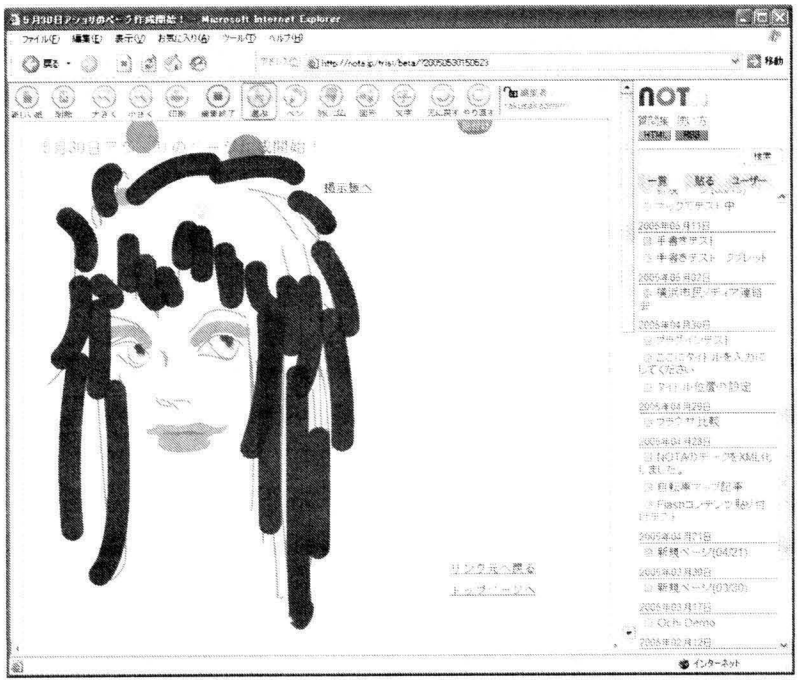

図 4.13　NOTA 上で、手書き線を用いてレタッチした例

表 4.2: 各種 Web ページ作成システムの基本機能の比較

	手作業管理方式	Wiki 方式	NOTA 方式
ページの作成	×動的な作成は不可。	○単語によるリンクもしくは新規作成ボタン。	○単語のリンクもしくは新規作成ボタン。
ページの作成	×動的な作成は不可。	○単語によるリンクもしくは新規作成ボタン。	○単語のリンクもしくは新規作成ボタン。
リンクの作成	×ファイル名による指定が必要。	△単語 (WikiWord) に対し自動でリンクが作成される。	○既存のページの一覧から選ぶとリンクが作成される。
書式の設定	◎ HTML の範囲内の書式すべて。	×定められた書式をタグで指定。	○サイズ、色など基本的なことのみ可。
画像の貼り付け	○画像ファイルをファイル名で指定。	△画像ファイルを別途アップロードし、ファイル名を指定。	◎画像をアップロードすれば、直ちにページに貼り付けられる。
手書きペン	×手書きで描いた画像を用意すれば、可。	×同左。	○ページ上にマウスのドラッグにより手書き入力可。
図形の描画	×図形編集した画像を用意すれば、可。	×同左。	○ページ上で簡易作図が行える。
リアルタイムの更新	×ローカルで編集したものを FTP でアップロードする。	△ Web 上で編集し、[送信] ボタンを押すと更新される。	◎変更があるたびに保存し、表示が更新される。
ブラウザ上での編集	×エディタ、ホームページ作成ソフトを用いて行う。	○ HTML のフォームを用いてブラウザ上で編集できる。	○ Flash Player を用いてブラウザ上で編集できる。
WYSIWYG な編集	○ホームページ作成ソフトの中には、WYSIWYG な編集をサポートしているものが多い。	×テキストベースで編集しなければならず、タグやフォーマットの知識が必要。	◎編集画面と出力結果は全く同一である。
ユーザーごとの権限の変更	×そもそも管理者一人しか編集できない。	△ページの凍結や、パスワードによる編集制限などができる。	○ユーザーごとに、どこまで編集できるか細かい設定が行える
互換性	○一般のブラウザで表示できる。	○同左。	△ Flash Player6.0 以上が導入されたブラウザが必要。

まとめ	ページの体裁などを思い通りに編集できるが、複数人によるリアルタイムの協同編集はできない。	グループコミュニケーションツールとして優れている点が多いが、凝った書式や、画像中心のページを作成するには不向きである。	複数人でリアルタイムに編集できる。図形や画像の貼りつけも可能でほぼ思い通りのページを作成できるが、Flash が導入されていない端末からは編集できない。

4.6　まとめ

現在、ホームページを作成・管理する際には、HTML の管理とサーバーへのアップロード作業が必須であるが、これは面倒で難解な作業であり、初心者のみならず、多くのユーザーの制作意欲を削いでいる。NOTA は、自由なレイアウト作成機能とリアルタイムの協同編集機能によって、Web 上で、グラフィカルなページを複数人で同時に編集するという新しい利用方法を可能にした。

第5章

アプリケーション・プラットフォームとしてのNOTA

NOTAに、アプリケーション・プラットフォームとしての機能を実装した。これにより、NOTAの利用方法を大きく変えるような変更をNOTAに加えることが可能になった。本章では、その具体的な仕様と、実現可能な機能について述べる。

5.1　はじめに

共生開発スタイルによる開発を促進するため、利用者の細かい要求にあわせてシステムをコミュニティ内部で改変できることが必要であると考え、NOTAにアプリケーション・プラットフォームとしての機能を実装した。これによって、NOTAの利用方法を大きく変えるような変更をNOTAに加えることが可能になった。NOTAのアプリケーション・プラットフォームを利用すれば、NOTA上で「NOTAプラグイン」と呼ばれるさまざまなFlashアプリケーションを動作させることができる。またそれらのプラグインは、「NOTA API」と呼ばれる一連のインタフェースを利用して、サーバーサイドに情報を蓄積し、それらの情報をクライアントで同期して利用するアプリケーションを容易に開発することができる環境を実現した。その具体的な仕様と、実現可能な機能について説明する。

5.2　プラグインの使い方

NOTAでは、ページにさまざまな機能を追加する「プラグイン機能」が利用できる。現在、NOTAには初期状態で、「アクセスカウンター」「簡易掲示板」「ツールバー色変更」「Webカメラ窓」プラグインが搭載されている。プラグインをNOTAのページに挿入する方法は以下の通りである。

1. 画面右側の「一覧」で「貼る」タブを選択する。
2. 「NOTAプラグイン」欄から貼り付けたいプラグインを選択する。

プラグインは、1ページにいくつでも貼り付けることができる（図5.1）。貼り付けられたプラグインは、図形や画像と同じように、左上の「移動つまみ」から好きな位置に移動できる。また他の部品と重ねて表示したり、プラグインの上から手書き線を描画することができる。不要になったプラグインは、選択して「×」ボタンを押すことで削除できる。これによって、単体としては高度ではないアプリケーションであっても、NOTAのプラットフォーム上で他の部品やプラグインと連携し、NOTAの用途を拡張する。

簡易掲示板プラグイン

簡易掲示板プラグイン（図5.2）を利用すれば、NOTA上で、掲示板の機能を使うことができる。「簡易掲示板」プラグインでは、管理者権限のユーザーのみが発言の削除を行うことができる。特定の発言を削除する場合には、名前の右横に表示されている「×」印部分をマウスで押す。

5.3　プラットフォームの実装

NOTA上でFlashアプリケーションを自由に動かすことができる機能を開発したことはすでに述べた。さらに、NOTA APIと呼ばれる一連の関数を利用すれば、サーバーとデータをやりとり

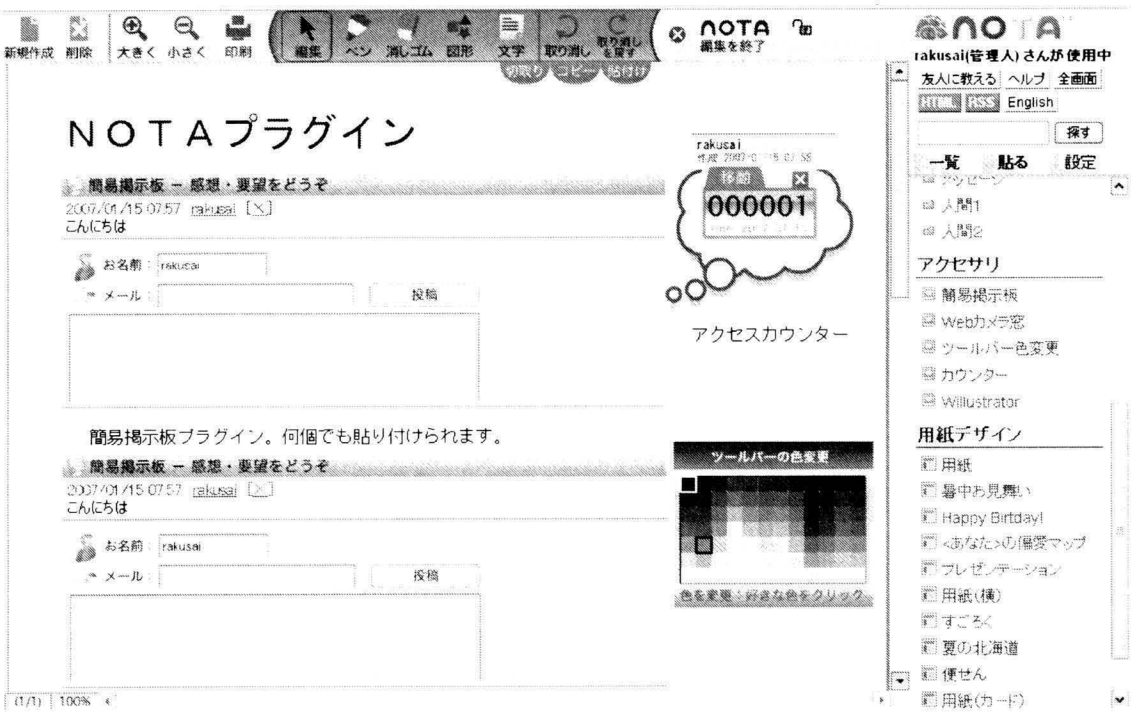

図 5.1　NOTA プラグインを貼り付けた画面例

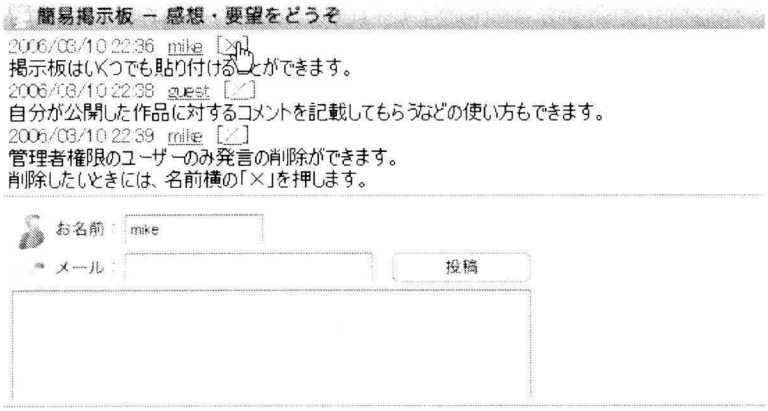

図 5.2　簡易掲示板プラグイン

する Flash アプリケーションを迅速に開発できる。またプラグイン同士を連携させることが可能である。Flash アプリケーションを開発することは、特にサーバーとのデータ通信において技術的に難しいが、NOTA ではこの通信部分が関数化 (NOTA API) されており、プラグインでもこれらの機能をそのまま利用できる。XML を用いたサーバーへのデータ保存、読み込みができる。複数人で情報を共有し、利用するための Flash アプリケーションを短時間で開発できる。

図 5.3　NOTA プラグインの例

5.3.1　NOTA API を用いた開発

NOTA プラグインは、サーバーの plugins フォルダに Flash ファイル（swf ファイル）を設置すれば、ブラウザ上で NOTA のメニューにプラグイン名が表示される。メニューをクリックするとページに配置することができるため、ユーザーが手軽に利用できる。

NOTA の API を呼び出すためには、Flash の Action Script の中で、NOTA API クラスを呼び出して利用する。

NotaAPI = new NotaAPI(this);

上記のように NOTA API のオブジェクトを呼び出す。引数には、NOTA のイベントを受け取る MovieClip を指定する。通常、「this」を入れる。この記述は、スクリプトのはじめの方で、プロジェクト 1 つにつき一度だけ呼び出す。

サーバーの保存場所

NOTA プラグインは、プラグインの部品ごとに固有の情報（プロパティ）をサーバーに保存しておくことができる。情報は XML 形式で、サーバーの NOTA データフォルダ内に「プラグインの ID 名.xml」という名前で保存される。プラグインは、NOTA が自動で決定するファイルとは異なるファイルにアクセスことはできない。しかし、下記の関数を用いることで、サーバーのファイルに関して意識することなく、情報を扱うことができる。

具体的なコーディング方法

具体的なコーディングの流れをソースコード 5.1 に示す。より詳細な情報に関しては、NOTA API マニュアル [永田]（図 5.4）に掲載している。

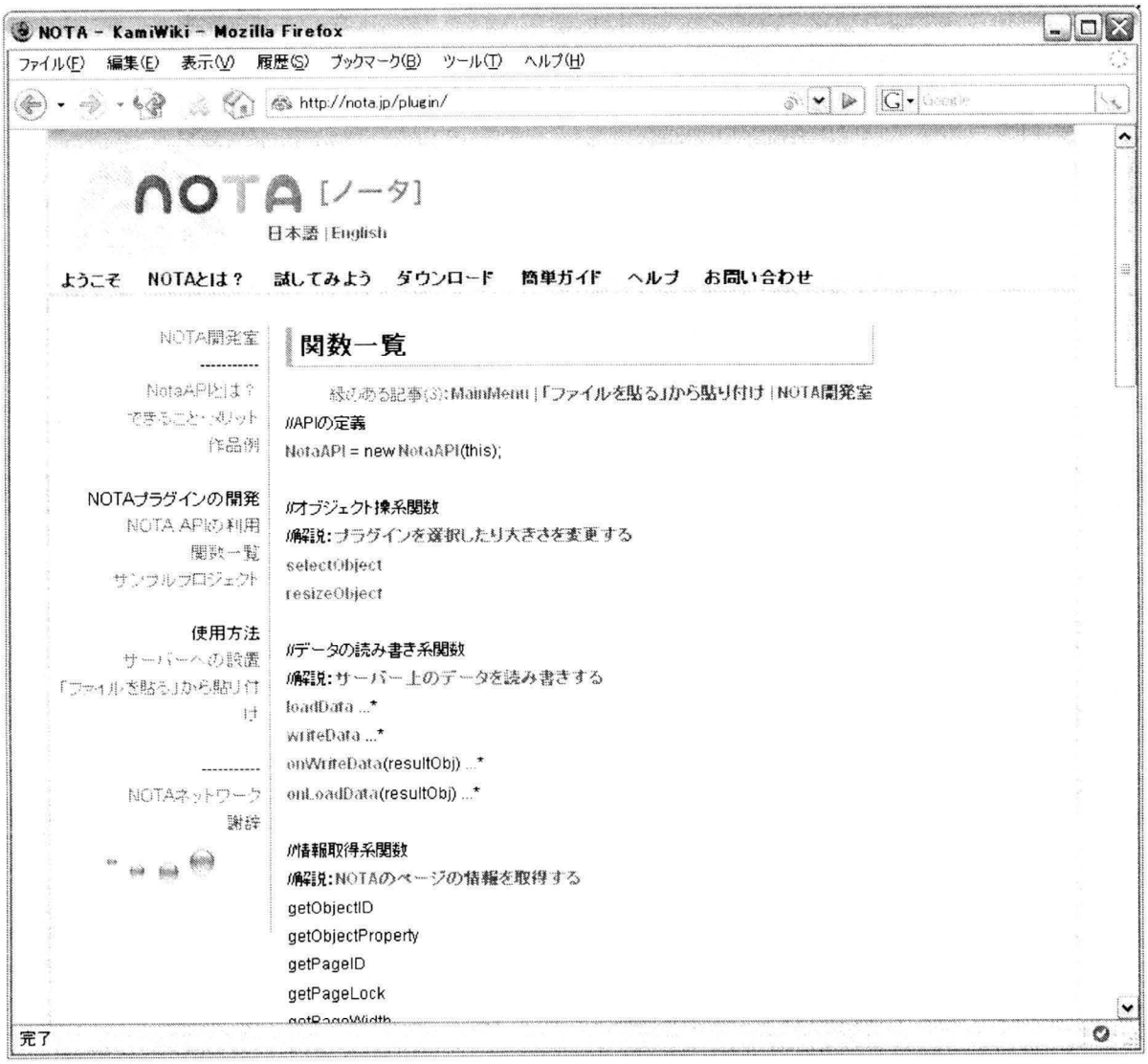

図 5.4　NOTA API マニュアル

ソースコード 5.1　NOTA API の利用例

```
1  //NOTA API を呼び出す
2  NotaAPI = new NotaAPI(this);
3
4  //サーバーからデータを読み込み開始
5  NotaAPI.loadData();
6
7  //読み込みが完了したら通知される
8  function onLoadData(resultObj){
9    //データが読み込まれた
10   var name = resultObj["default.name"];    //名前の取得
```

```
11    var email = resultObj["default.email"];    //メールの取得
12    //ここで、適当な処理を行う
13  }
14
15  //データをサーバーに書き込む
16  var newObj = new Object;
17  newObj["default.name"] = name.text     //名前のセット
18  newObj["default.email"] = email.text;    //メールのセット
19  NotaAPI.writeData(newObj);
20
21  //サーバーの書き込みの結果が通知される
22  function onWriteData(resultObj){
23    //書き込み結果
24    if (!resultObj){
25      MessageBox("掲示板：書き込みに失敗しました。");
26      return;
27    }else{
28      //書き込み成功
29    }
30  }
```

NOTA APIの使用例

例えば、アクセスカウンターのプラグインを作ることを想定する。アクセスカウンターは、サーバーに現在のアクセス回数が記録されており、ブラウザでページが読み込まれるたびに、その回数を取得し、数字を1足して、またサーバーに保存する。この場合、NOTA APIを利用してloadData関数を呼び出し、onLoadDataイベントで受信した値をMovieClipに表示させ、その後、値を1足し、writeData関数を呼び出す。

5.3.2　NOTA APIを利用した作品例

NOTAプラグインで、考えられる作品は以下に列挙する。NOTAのプラグインを開発するときは、常にリアルタイムに複数人が使用することを考慮する必要がある。また、NOTA APIを使って、リアルタイム性を活かした開発が可能である。

NOTAプラグインで開発できる作品例

- アンケート、掲示板など
  多数のユーザーの情報を集めるもの
- ストリーミング通信
  ビデオチャット、ストリーミング放送など動画を用いたもの
- インタラクティブゲーム
  さまざまな楽しいゲーム
- 簡易ボタン
  押すと他のサイトにジャンプできるアーティスティックなボタン
- 表作成
  表を作成したり、計算できる
- ショッピングカート
  インタラクティブなショッピングカート
- ツールバー
  自分好みにデザインを変えたツールバー

5.4　まとめ

NOTAのアプリケーション・プラットフォーム機能の具体的な仕様と、実現可能な機能について述べた。本機能は、プラグインをユーザーが必要なときにクリックしてページの好きな位置に貼り付けて使うという新しい利用方法を持っていることを説明した。また、サーバーとの通信を簡略に実装できるNOTA APIと呼ばれるインタフェースを利用して、複数のユーザーが同時に情報をやりとりするような機能を持つプラグインを開発可能であることを述べた。

第6章

NOTAの利用事例と応用

NOTAネットワークの活動により、NOTAは横浜市を中心に多くの市民団体に普及した。本章では、それらのNOTAを活用したコミュニティのうちいくつかを取り上げ、NOTAの利用事例を述べる。

6.1　はじめに

NOTA は、自由度の高いプラットフォーム指向のシステムであり、市民などの一般ユーザーを対象に設計されていることは前章までにすでに述べた。

2004 年以降は、NOTA ネットワークの活動により、NOTA は横浜市を中心に多くの市民団体に普及した。本章では、それらの NOTA を活用したコミュニティのうちいくつかを取り上げ、NOTA の利用事例を紹介する。

ここでは、NOTA の使い方を工夫してコミュニティ活動に合わせて利用した事例と、コミュニティ内で NOTA のプラグインを開発して NOTA を拡張して利用した事例の二つに分けて論じる。

6.2　NOTA の使い方を工夫した事例

NOTA の使い方を工夫してコミュニティ活動に合わせて利用した事例について述べる。

6.2.1　サバイバルジュニアプロジェクト

小学 3 年〜6 年までの子供たちが防災について学び、地域の防災情報を調査するプロジェクトにおいて NOTA が利用された。本プロジェクトは、地域の防災情報を学校外の活動として大学、地域が連携して実施したものである。

2005 年 8 月〜9 月にかけて行われた防災キャンプとシンポジウムにおいて NOTA が用いられた。NOTA を自分の住む地域の防災情報を調べたことを書き込み、発表する場として使っている。それぞれの地図は子供が自分で歩いて調べ、NOTA に書き込んだ。これらの情報は NOTA を使うことによって、いつでも新しい情報を自分たちが更新することができる。

6.2.2　つづき Traffic Project

「つづき Traffic Project」では、横浜市都筑区における道路や交通の状況を市民が NOTA 上で報告し、そのページを市に提出することによって実際にその場所の道路の改善につなげるという試みが行われた。交差点ごとに NOTA のページを作成し、そのページに交差点の写真がアップロードされている。その写真に注釈をつけたり、場所に関しての議論がそのページ上で展開される。各交差点のページへは、市全体の地図をアップロードしたページからリンクが張られ、地図上の地点をクリックすることで、移動できるようになっている。写真や手書き文字などのリッチコンテンツが Web 上で書き込める点と、誰でも参加できる点が有効に利用された。この NOTA のページを市に提出し、実際に道路が改善されるという成果も得られた。NOTA は最初からリッチコンテンツを作成できるがゆえに、行政に提出する際に改めてデザインを作り直す必要がない。そのため、説得力のある文書をすぐに作成でき、結果的に市民の意見を政策に反映するまでの時間を短縮できた可能性がある。このプロジェクトは現在も進行中である。

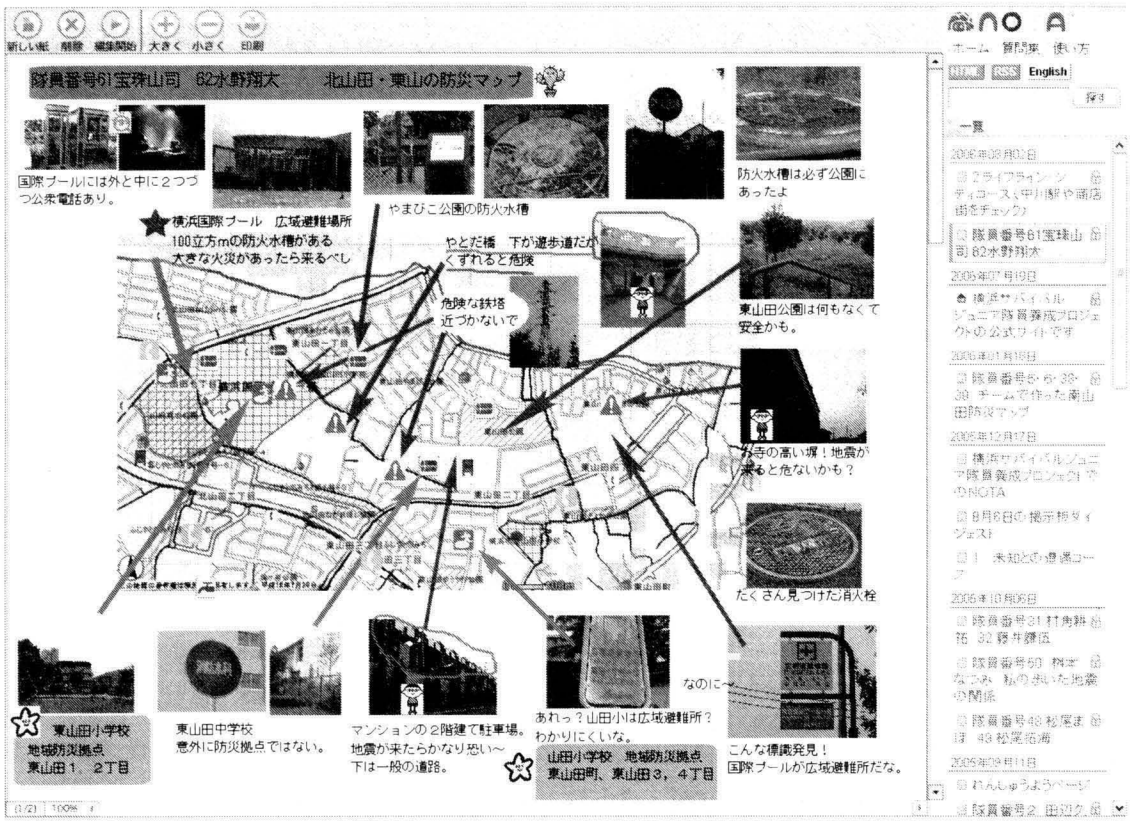

図 6.1　サバイバルジュニアプロジェクトの作品例

6.3　NOTA のプラグインを開発した事例

コミュニティ内で NOTA プラグインを開発して NOTA を拡張して利用した事例について検討する。

6.3.1　障害者コミュニケーション支援プラグイン

障害者コミュニケーション支援プラグインは、京都の養護学校において、障害者を持つ生徒がアイコンで意志を伝えるために制作された。画面例を図 6.4 に示す。ユーザーは、伝えたい内容が描かれたアイコンを選択し、NOTA のページにアイコンを貼り付ける。アイコンはプラグインとして制作されており、クリックするとアイコンの意味を日本語の音声で再生する。NOTA 上でプラグインを自由に移動できるという機能を利用して、ユーザーは、アイコンを並べ替えたり、サイズを変更したりすることによってより複雑な意志を伝えることが可能になっている。

図 6.2　つづき Traffic Project の表ページ

6.3.2　NOTA ビデオ会議室プラグイン

NOTA のプラグインとして、ビデオ会議室プラグインが開発された（図 6.5）。開発は NOTA ネットワークのメーリングリストを通じて千葉恭弘氏を中心に行われた。このプラグインを利用すると、NOTA のページ上に会議室プラグインを貼り付けるだけで、多地点の Web カメラの映像を同時にストリーム配信することができる。音声にも対応しており、マイクを使用することで、多地点の音声の相互配信ができる。本システムの新規性は、クライアントに特殊なソフトウェアをインストールする必要がない点にある。また、NOTA 上で動作するため、ビデオ会議室をしながら、同時に図を利用した情報共有が可能である。普及レベルにあるビデオ会議室システムとして Skype と MSN Messenger があるが、多地点の映像の相互配信はサポートしていない。ビデオ会議室プラグインでは、多地点の映像の相互配信に加え、不特定多数のクライアントに映像を配信することができる。つまり、会議の「聴講」が可能である。

図 6.3　つづき Traffic Project の詳細ページ

実装

サーバー側は Flash Media Server のフリー版を利用した。システムは、Flash アプリケーションとして開発し、NOTA API を利用して NOTA プラグインとして動作させている。

運用

2006 年 1 月 11 日に開催された NOTA ネットワーク関西のミーティングにおいて、発表の様子を NOTA ページで 13 時から 18 時まで 5 時間連続してストリーミング配信した。合計 4 つのクライアントがこのページにアクセスし、会場の映像と音声を受信した。1 つのクライアントでは、一定間隔でブラウザがダウンするという不具合が起きたが、それ以外のクライアントでは正常に受信できることが確認できた。

図 6.4　障害者コミュニケーション支援システムの利用例

6.3.3　C-Shirts プラグイン

C-Shirts は、NOTA を使って、CreativeCommons ライセンスの作品を組み合わせて T シャツのデザインを作り、それを実際の T シャツ（C シャツ）として注文することもできるシステムである。NOTA 上で T シャツの枠を表示させ、注文ボタンを押すと、作成したデザインを画像に変換し、購入サイトに移動する。C シャツには、QR コードで NOTA の URL が入っているので、街中で見かけた C-Shirts の URL を取得すると、そのデザインをもとにリミックス（再編集）して、自分の C-Shirts を作ることもできる（図 6.7）。

6.4　利用経験

ここではコミュニティにおける運用によって得られた経験について述べる。運用を通じて、通常の CGI を用いた動的作成 Web とは異なる、NOTA ならではの使われ方が分かってきた。NOTA の特徴的な利用経験を以下にまとめる。

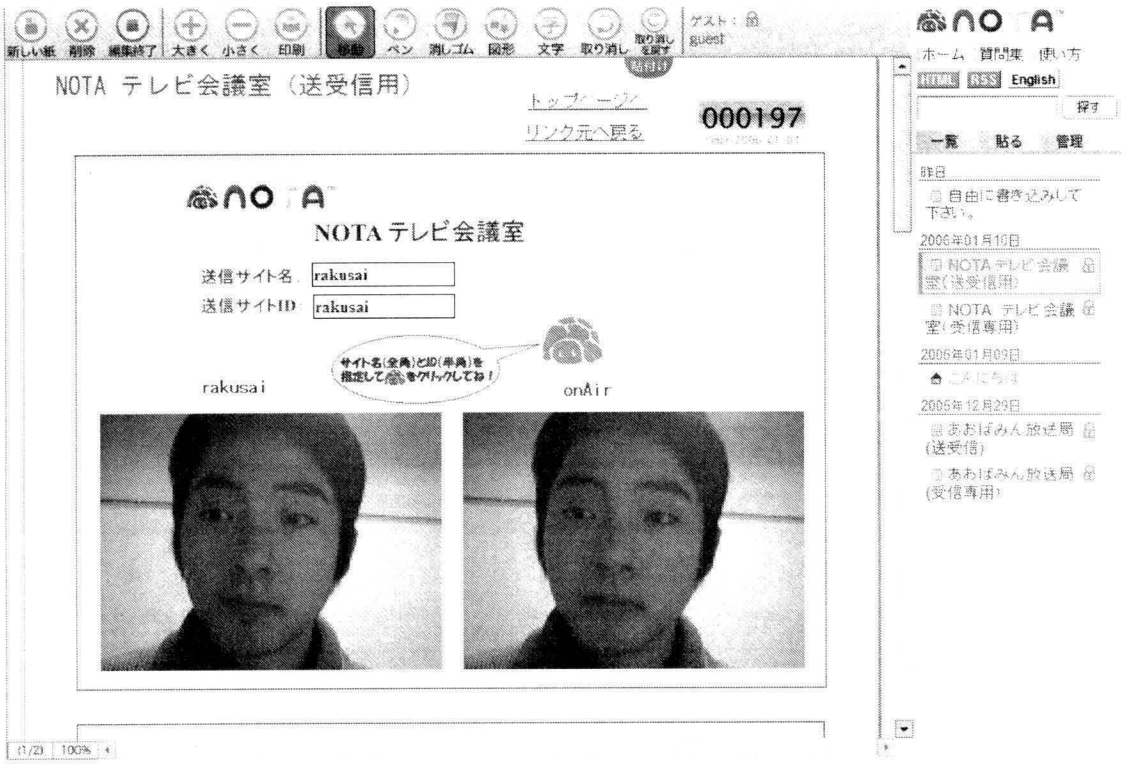

図 6.5　NOTA ビデオ会議室プラグインの利用例

画像のアップロード

画像を手軽にアップロードしページに貼り付けられることは、すでに見てきたように NOTA の特有の機能であるが、この機能の積極的な活用を確認できた。1 ページに 10 個の画像を貼ることはよくあり、中 20、30 個の画像を貼るユーザーもいた。また、貼り付けた画像のサイズや配置はたいていのユーザーが貼り付けた後で見やすいように調整しているのが確認できた。NOTA のオブジェクトの自由配置の機能が頻繁に使われ、活用されていたといえる。

他人の部品への注釈

他人が作成したテキストや写真に対し、手書き入力機能を用いて、積極的に線を引く用いられ方が頻繁に見られた。また他人の部品の隣にテキストを配置し、注釈を書く例もよく見られた。それを参考に元のテキストを修正したり、その場で議論をするなど NOTA ならではの使い方が違和感なく受け入れられたようである。ただ、注釈があまりに多くなると、ページの視認性が下がり、結果としてページの本筋が分かりにくくなることもあった。ページの内容を表示する場と議論をする場が分かれていない NOTA 特有の問題である。

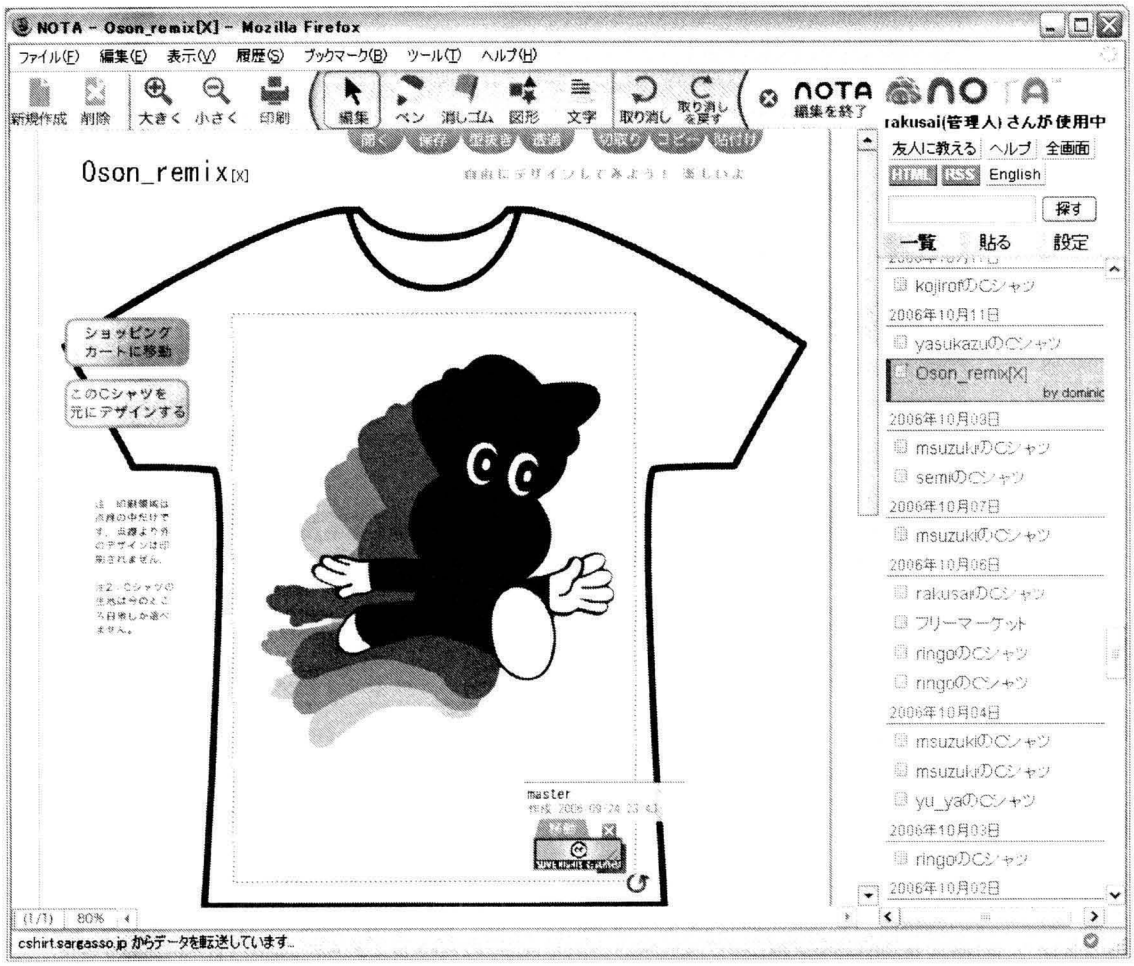

図 6.6　C-Shirts の作例

チャット

NOTA の書き込みの手軽さと、リアルタイム性によりページ上でチャットの代わりとして用いられることもあった。話題の遠近により、文字の配置場所を変えるなどの工夫をユーザーが自主的に行っていることが多く、単に話題を時間軸で並べただけのチャットと比較して話題を整理し、活用するするのに有効であることが分かった。

ゲスト投稿

NOTA はページの編集が可能なゲスト権限を用意し、不特定多数のユーザーの書き込みを認めているため、第三者によってページの趣旨に沿わない書き込みが投稿され、トラブルになるといった危険性を孕んでいる。しかしながら、ゲストが既存のオブジェクトを削除できず、追加書き込みだけが可能である点と、管理者とコミュニティのメンバーがそれらの投稿を発見次第、削除できるため、コミュニティのメンバーが常に監視しているページにあっては、これらのトラブルは起きて

図 6.7　C-Shirts の QR コードを携帯電話を利用して読み込む様子

いない。

6.5　まとめ

NOTA の利用事例として、NOTA の使い方を工夫してコミュニティ活動に合わせて利用した事例と、コミュニティ内で NOTA のプラグインを開発して NOTA を拡張して利用した事例についてそれぞれ紹介した。また、NOTA の運用から分かった知見をまとめた。

第 7 章

関連研究

本章では、本研究と関連を持つ先行研究を挙げ、それらの特徴や本研究との相違点、および本研究の位置づけについて述べる。

7.1　はじめに

本章では、本研究と関連を持つ先行研究を挙げ、本研究との関連や差異を述べ、本研究の位置づけを明らかにする。本研究は共生開発スタイルのコンセプトおよびシステムの両方を研究対象としているため、関連研究もそれに沿った形で述べる。関連研究のうち、第 2 章で挙げたものは、本章では省略する。

7.2　Web 技術の新領域

技術的な背景には、近年の Web の進化がある。Web2.0[梅田 06] というキーワードに総称されるように、Web は、「非同期通信」の技術や「Web の API」が普及し、リッチ・インタフェースの開発や、外部のシステムと連携する機能の開発が盛んになった。これらの Web の新しい技術的な概念について述べる。

7.2.1　非同期通信

ウェブブラウザ上で画面遷移することなく、情報をサーバーに対して送信したり取得したりすることを Web の非同期通信と呼ぶ。実装には、JavaScript[*1]や Adobe Flash を用いる。従来は、Web において動的なコンテンツを生成するときは、ページが遷移したときに POST もしくは GET 方式で情報をサーバーに送信し、コンテンツを書き換えていた。非同期通信により、ウェブブラウザ上で既存の枠組みにとらわれないユーザーインタフェースを実現することができる。代表的な例である Google Maps を図 7.1 に示す。Google Maps では、ユーザーが地図をマウスでドラッグすると、システムは、画面の枠の外にある地図情報を逐次サーバーから取得して描画するため、地図をスムーズにスクロールすることができる。

NOTA においては、複数人が同時に 1 つのドキュメントに対して NOTA 上で書き込みを行うときに、非同期通信を利用して、同一の画面を共有できるようにしている。

7.2.2　Web の API

Web の API（Application Program Interface）は、用途によって「外部のシステムとの連携」と「プラグインによるシステムの拡張」の二つに分類される。

外部のシステムとの連携

従来、Web を利用したシステムは、1 つのドメインにおいて単独でシステムを運用することが多かったが、近年は、外部のシステムと連携するために、積極的に情報の相互利用を行うものが増えてきた。これらを実現するために、ドメインを超えて他のシステムの情報を取得したり、送信し

[*1] JavaScript を用いて非同期通信を行うことを特に「Ajax（Asynchronous JavaScript + XML の略称）」と呼ぶ。

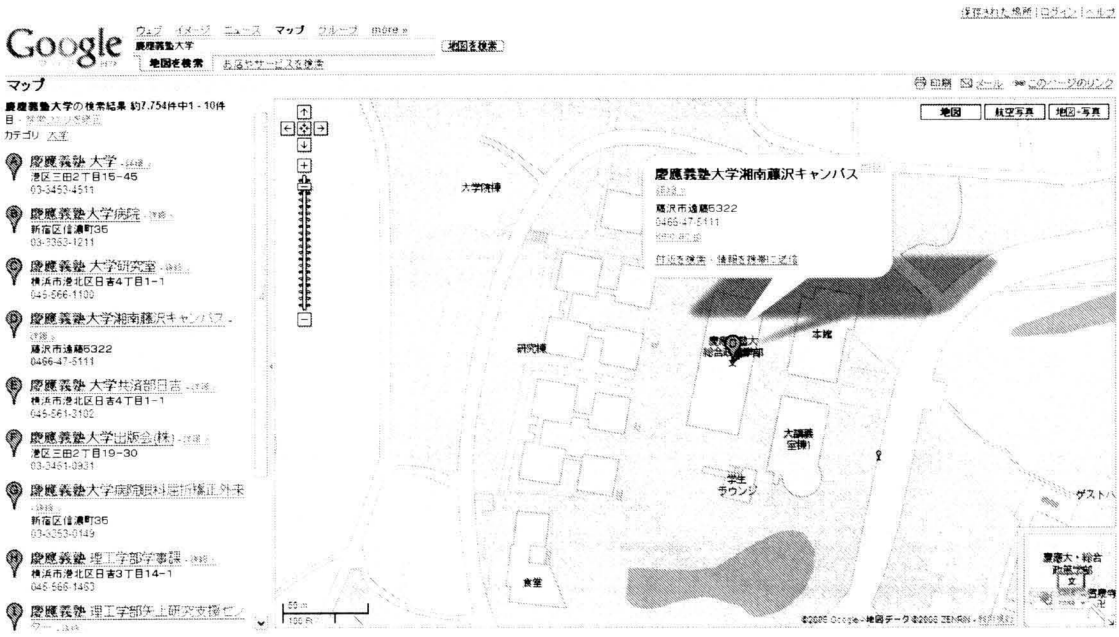

図 7.1　Google Maps

たりする際に利用可能な関数やプロトコルの集合を Web の API と呼ぶ。

プラグインによるシステムの拡張

また、ウェブシステムの中には、そのウェブシステム自体を拡張することが可能なプログラム（プラグインと呼ばれる）を、外部の開発者が追加することができるものもある。この場合も、そのシステム上で多くのプラグインが共通して利用する機能は、API として提供される。

NOTA においても、NOTA を拡張するプログラムに対して、NOTA API と呼ばれる一連の関数群を提供し、コミュニティ支援のために有効な機能を開発しやすくなっている（詳しくは、第 5 章で述べる）。

7.3　コンテンツ流通の新領域

7.3.1　オープンコンテンツ

Web の技術的な進化により、コンテンツの流通や利用にも変化が起きている。Web のシステム間の連携が容易になったため、従来は 1 つのシステムに蓄積されていコンテンツがシステムを超えて流通するようになった。また、Web のユーザーインタフェースの進化により、Web 上での直接コンテンツの作成、編集、複製ができるようになり、コンテンツの利用形態が変化した。このようにコンテンツが Web 上で作成、編集され、システムを超えて流通する現象を、「オープンコンテンツ」と呼ぶ。また、システムがオープンコンテンツの作成や流通をサポートすることを、ADOC[鈴木 06]（Application Driven Open Contents、アプリケーション駆動型オープンコンテ

図 7.2　Willustrator

ンツ）と呼ぶ。NOTA も、ADOC の 1 つとして捉えることができる。

Willustrator

Willustrator[神原 06] は、「Web 上での協同イラストレーション」というコンセプトの元に作られた Web のドローツールである。複数ユーザーで利用できるウェブアプリケーションであり、ブラウザ上で利用できるドロー UI を持つ（図 7.2）。ペイントツールと違い、ドローツールであることを活かした再編集機能や他のユーザーの絵を簡単に再利用するための仕組みを持っている。Willustrator のドロー機能は NOTA より細かな制御が可能である。しかし、制作した絵を他のツールやコンテンツに埋め込んで利用することを想定しており、NOTA のようにウェブページを作ることはできない。また、プラグインによって拡張し、アプリケーション・プラットフォームとして利用する機能はない。NOTA は、文字や写真、図、プラグインなどを貼り付けるプラットフォームを指向している。

7.3.2 Consumer Generated Media

ウェブサイトのコンテンツを、従来のようなプロのコンテンツ制作者（作家、記者、プロデューサーなど）ではなく、エンドユーザーが作成し提供することを、Consumer Generated Media (CGM と略す）と呼ぶ。Social Bookmark や、ウェブログ、SNS などはその代表的な例である。NOTA も、CGM の 1 つである。

7.3.3 Creative Commons

Creative Commons[Les00][Les01][L.02] とは、さまざまなコンテンツに対して著作権を保持しながら一定の自由を事前に許諾していることを、分かりやすく表示することができるライセンス群のことである。例えば、あるアーティストの曲をミックスして新たな作品を作りたいという場合、従来は、作者と管理団体に説明、交渉、契約準備、確認といったことが必要であった。しかし、Creative Commons のライセンスを適用していれば、このような煩雑な手続きは必要ない。

Creative Commons に対応するためには、コンテンツを公開する際にライセンスの条件のマークを選択し、表示しておくだけでよい。ライセンスの条件のマークには、例えば、原著作者の「氏名表示」が必要かどうか、非営利の活動のみに使用可能かどうか、原著作者と同じライセンスで発表しなければならないか、表現やソフトウェアを改変して用いていいかどうかといったものがある。

Creative Commons は、ライセンスの提供だけではなく、ライセンスをシステムが識別するための仕様の提案も行っている。例えば、HTML ファイルや画像ファイルに対して、Creative Commons が規定する方法でライセンスを記述することによって、Google や Flickr から Creative Commons に対応したコンテンツを検索することができる。

Creative Commons により、7.3.1 に示したオープンコンテンツの流通の法的、技術的な煩雑さが解消されることが期待される。NOTA も、複数人で協同作成したコンテンツのライセンスに関して複雑な課題を抱えており、こうしたコンテンツの新しいライセンス方式と深く関係している。

7.4 開発形態に関する関連研究

7.4.1 CSCL および CSCW

NOTA のようにコミュニティ活動と持続的な関わりを持ち、開発者が市民の中から現れ、利用者と一体となって設計し開発するという視点は、従来の CSCL (Computer Supported Collaborative Learning、コンピュータに支援された協同学習)、CSCW（Computer Supported Collaborative Work、コンピュータに支援された協同作業）研究にもなかったものである。こうした研究にユーザーの実践を見た上でデザインするという観点はあったにしても、設計者（開発者・デザイナー）は、ユーザーを個々に分解し、一人のユーザーの視点から使い勝手や効果を評価していたにすぎなかった。つまり、システムの設計において、設計者と利用者を含めたネットワーク構築のあり方に

焦点を当てたものはほとんどなかった。[*2]

7.4.2　参加デザイン

参加デザイン（Participatory Design）のアプローチはどうだっただろうか（[BG96] [KB98] [KSB98]）。参加デザインのアプローチは、ワークプレイスにおけるシステム、人工物の設計への「ユーザー」の参加をめざしたものであり、ワークショップの企画やユーザーも含めたデザイン場面を組織化するといった実践を行ってきた。しかし、こうした場面設定は一時的なものであり、長期的に関係を持続する設計者と使用者のネットワーク構築をめざすものではない。さらに、設計者と利用者は明確に区分されており、利用者が設計をサポートすることはない。このアプローチでは、設計者は、ユーザーのニーズや注文を聞くが、実際の開発は、あくまで設計者が行っていた。もちろん、実際に参加デザインを組織化するためには、コミュニティが必要となるが、参加デザインは、こうしたことに焦点を当ててこなかった。

NPO パンゲア

NPO パンゲア [高崎 06][NPO パ] は、「世界の子供たちをつなぐ」というコンセプトを持ち、「絵文字コミュニケーター」、「コエツナ」、「パンゲアネット」など複数のソフトウェアを開発し、それらのソフトウェアを利用して世界中で「アクティビティ」と呼ばれる交流活動を行っている。ソフトウェアは、一部の常任メンバーを中心に、多くはボランティアによって開発されている。ソフトウェアの開発は、「まずは動くことを重視」「プログラミング言語はボランティアの得意な言語で開発するためさまざまな言語が入り交じる」「ネットのセキュリティには配慮」「アドバイスをいただいた後、即反映させるか待つかどうかは NPO の実証活動フェーズも含めた全体俯瞰の上、担当責任者が決定する」といったプロセスで開発を進める。NOTA のコミュニティと同様、開発者は NPO のコミュニティメンバーである国内外の多様な人たちとコミュニケーションをしながら開発する。それには、小中学生、小中学校の先生、ベンチャー企業人、大企業役員、保護者、デザイナー、記者、研究者、ミュージシャン、学生、官僚、フリーターなどが含まれる。開発には、モチベーションが重要視される。ボランティアは、モチベーションが低い人は、もともと参加しないため、金銭的なインセンティブが低く、開発者モチベーションが極めて高いという特徴を持つ。モチベーションは、新規開発時には、「楽しい、社会的意義を感じる、誇らしい」などの感情によって高まり、デバッグ時には、「恥ずかしい、教われて勉強になる、必要性を実感する」などの感情よって保たれる。 NPO パンゲアでは、(1) プロジェクトのミッションをぶらさない、(2) ソフトウェア品質を人/環境のせいにしない、(3) ソフトウェア職人を惹きつける、(4) 感動できる感性をキープする、などを組織的に重視している。NPO パンゲアの開発形態は、本研究で提案する「共生開発スタイル」と類似している。開発者がコミュニティの中で多様な人たちと協働しながら開発を進める点や、ボランタリーな動機を重視している点が共通している。その一方で、NPO パ

[*2] ニューマン [New98] は、アクターネット的な観点を用いながら、あるミドルウェアのデザインの事例を分析し、システムをデザインすることが、さまざまな技術や人々のネットワークを構築することに等しいということを示している。

ンゲアでは、常任のリーダーが開発を指揮を執っているのに対し、NOTA の事例においては、明確な開発のリーダーが存在しないという点が異なっている。

第8章

共生開発スタイルの考察と今後の展望

本章では、共生開発スタイルについて、現在分かっている課題や限界、今後の展望について述べ、本研究のさらなる発展について論述する。

8.1　共生開発スタイルの考察

8.1.1　オープンソースからオープンコンテンツへ

すでに述べた通り、私が開発した NOTA は、開発者以外の人も参加する地域コミュニティの中で開発が進められ、改良が加えられた。このようにボランタリーな動機から開発が進められるという点は、オープンソース・ソフトウェアの開発と類似点がある。

近年、他人のコンテンツを流用し、そこに自分の著作を追加して公開するというオープンコンテンツの現象が注目されている。オープンソースの開発コミュニティにおいては「ソースコード」をオープンにして複数の人が少しずつ創作に貢献することでソフトウェアを開発する手法が定着しているが [Sta02]、一般のコンテンツでは、技術的な課題や法整備の遅れなどから、このような創作手法は一部を除いてあまり行われていない。

それらを解決するため、一般のコンテンツをオープンソース的に生成、伝播させることができるシステムが必要であると考える。NOTA（第 4 章で詳述）はこのためのプラットフォームを提供する。オープンコンテンツの実現には、コンテンツに多人数で同時に編集可能であることと、コンテンツを他人が容易に再利用可能であることが必要である。オープンコンテンツを実装するアプリケーションの設計をまとめると、以下のようになる。

オープンコンテンツをサポートするシステム（ADOC）の設計

1. 多人数で同時に協同編集できるシステムの設計
2. コンテンツの伝播と派生システムの設計

横浜市のコミュニティにおいて、共生開発スタイルの手法が成立した背景として、NOTA がこれらのオープンコンテンツの要件を満たしていたという点は重要である。専門的なソースコードではなく、文書、写真、アイコン、プラグインといったコンテンツの協同編集と派生をシステムがサポートしたことによって、開発者以外の人が開発コミュニティに参加することができたといえる。

8.2　今後の展望

共生開発スタイルは、地域コミュニティにおいて必要な機能をコミュニティ内で開発できるという点で利点がある一方で、課題もある。

1 つ目は、プラグインなどをコーディングできる開発者がまだ限定的なものにとどまっている点である。Flash や NOTA API を利用した開発は、従来の開発と比べて敷居が下がっているが、誰もが使えるとはいえない。Flash は、コーディングをしなくても条件や関数を選択して組み合わせることで簡易的なスクリプトを作成できるが、今後はこのようにコーディングという行為をすることなく開発ができるような工夫をしていく必要があるだろう。

二つ目は、共生開発スタイルによって開発されたソフトウェアは、汎用性が低くなりがちになる

点である。そもそもコミュニティにおける狭いニーズに応えるために開発するため、開発されたソフトウェアは、同じような条件のコミュニティにおいてしか有効でないことが多い。汎用性が低くなると、ソフトウェアの蓄積がなされず、どのコミュニティにおいても似たような機能を再生産することになる。この課題は、ユーザーが作成したシステムから汎用性のある機能を抽出し、蓄積できるようなフレームワークを考案することで解決できる可能性がある。

今後は、これらの課題を解決するために継続的に研究に取り組んでいく予定である。

第 9 章

結論

本章では本論文を総括する。

9.1　研究の成果

本研究の成果は、以下の 4 つにまとめることができる。

- 市民が地域コミュニティにおいて開発を行う「共生開発スタイル」を提案した。
- コミュニティ活動支援システム「NOTA」を開発した。
- NOTA を拡張するアプリケーション・プラットフォームを実現した。
- NOTA の利用事例を評価し、有効性を検証した。

まず、市民がソフトウェアを開発し、コミュニティで運用する中で得られるフィードバックを元に継続的に改良していく「共生開発スタイル」を提案した。共生開発スタイルは、以下のようなメリットがあることを示した。

1. 効率をあまり考えずにシステムを継続して開発できる。
2. 利用者のフィードバックがただちに得られる。
3. 開発者とコミュニティが一体になっているため必要な機能だけを作ればよい。
4. 高いモチベーションを保って楽しく開発できる。

次に、共生開発スタイルの事例として私が開発した「NOTA」を利用して共生開発スタイルが広がったプロセスを明らかにした。NOTA は 2005 年より横浜市のいくつかの地域コミュニティで利用され、その後 NOTA を改良したいという要望を持つ利用者の有志が集まって開発コミュニティを形成し、共生開発スタイルが始まった。

NOTA は、複数の人が同時にブラウザ上で文字や図、写真などを好きな位置に書き込み、蓄積できるシステムである。NOTA の実装について述べ、汎用性の高いインタフェースを持つシステムであることを説明した。

共生開発スタイルによる開発を促進するために、利用者が NOTA を拡張できるプラグイン機能を追加した。この結果、NOTA を応用した地図システムや、ビデオ会議システム、障害者コミュニケーションシステムなどが地域コミュニティ内で開発された。これらの事例は、コミュニティの細かな要求に応える機能を的確かつ迅速に開発することに成功しており、共生開発スタイルの有効性を示したといえる。

9.2　論文の総括と結論

市民が地域コミュニティ活動のために自らソフトウェアを開発するという「共生開発スタイル」は、今後、その重要性が高まると考える。本研究は、工業的なソフトウェア開発やオープンソース・コミュニティにおける開発の限界を指摘し、Web の汎用的な情報共有システム「NOTA」を利用して地域コミュニティにおいて共生開発スタイルが成り立つことを示した。共生開発スタイルが広がることによって、地域コミュニティで必要とされる細かな要求を満たすソフトウェアがより多く開発されることを願う。

謝辞

慶應義塾大学大学院入学以来、研究をご指導をいただいた慶應義塾大学 安村通晃教授に深く感謝致します。また、本研究の副査として、ご意見、ご助言をいただいた慶應義塾大学 徳田英幸教授、横浜の地域コミュニティを紹介していただき、コミュニティの理論と実践の両面においてご指導いただいた武蔵工業大学 上野直樹教授に感謝致します。

研究の進め方や論文の構成について細部にわたって助言をいただいたインタラクション・デザイン・プロジェクトの樋口文人先生、渡邊恵太さんに感謝します。また、同期として共に励まし合った赤池輝幸君、後藤孝行君、佐藤公宣君、ありがとう。

本研究を通して、改めて開発コミュニティの大切さ、有り難さを実感致しました。NOTA の研究は、多くの方の献身的な協力によって成り立っています。さまざまな形で開発を支援してくださったみなさんにこの場を借りて深く感謝申し上げます。NOTA の開発を協力していただき、簡易セットアッププログラムやセットアップマニュアルを書いていただいた慶應義塾大学 秋元慶太さん、テンプレートイラストやロゴを提供していただいた武蔵工業大学 真行寺由郎さん、テレビ会議室プラグインを開発した他、さまざまな開発の助言をいただいた千葉恭弘さん、ありがとうございました。NOTA ネットワークのメーリングリストで、多くの方の技術的な質問に答えてくださった王子全主さん、國松基さん、NOTA の分かりやすいマニュアルを作成してくださった檜垣健太郎さん、横浜市のコミュニティのために無償で NOTA のサーバーを提供してくださった藤田幸子さん、NOTA のワークショップなどで指導にあたっていただいた川辺さつきさん、桜井得倫さん、ありがとうございました。最後に、ここでお名前を挙げることはできませんが、NOTA ネットワークのメンバー全員に感謝致します。みなさんのおかげで、この論文を完成させることができました。

本研究に関する発表

書籍

[1] 上野直樹，ソーヤーりえこ，永田周一：「学習環境のデザインのためのネットワーク指向アプローチ」．科学技術実践のフィールドワーク．pp.56-74．せりか書房．Dec, 2006.

学会講演会・シンポジウム

[1] 永田周一：なめらかなファイル操作体系の構築と実用化．平成 15 年度未踏ソフトウェア創造事業成果報告書．Feb, 2004.

[2] 永田周一：NOTA - Web の新しいコミュニケーションツール．第 12 回インタラクティブシステムとソフトウェアに関するワークショップ（WISS 2004）．Dec, 2004.

[3] 永田周一：市民メディアにおける新しいネットツールの活用．第 4 回 市民メディア全国交流集会．Sep, 2006.

[4] 永田周一：NOTA: あたらしいウェブのカタチ．第三回 Creative Commons Japan セミナー．Sep, 2006.

[5] 高崎俊之，永田周一，安川直樹：パネル「共生プログラミングが情報社会を救う」．情報社会のデザインシンポジウム 2006．Dec, 2006.

参考文献

[Bea] Kent Beck and et. al. Manifesto for agile software development. http://www.agilemanifesto.org/.

[Bec99] K. Beck. *Extreme Programming Explained: Embrace Change.* Addison-Wesley, Boston, 1999.

[BG96] S. Bodker and K. Gronbak. Users and designers in mutual activity: An analysis of cooperative activities in system design. In *Cambridge University Press: New York*, Vol. 2, pp. 130–158. Cambridge University Press: New York, 1996.

[BKC01] P. B. Bailey, J.A. Konstan, and J.V. Carlis. Demais: Designing multimedia applications with interactive storyboards. In *Proceedings of ACM Multimedia 2001*, Vol. 5, pp. 241–250, 2001.

[BT04] B. Boehm and R. Turner. *Balancing Agility and Discipline: A Guide for the Perplexed.* Addison-Wesley, Boston, 2004.

[Cal03] M. Callon. The role of hybrid communities and socio-technical arrangements in the participatory design. In *Journal of the center for information studies*, Vol. 5, pp. 3–10, 2003.

[CL97] M. Callon and J. Law. After the individual in society: Lessons on collectivity from science, technology and society. In *Canadian journal of sociology*, Vol. 2, pp. 165–182, 1997.

[HTOH96] K. Harada, E. Tanaka, R. Ogawa, and Y. Hara. Anecdote: A multimedia storyboarding system with seamless authoring support. In *Proceedings of ACM International Multimedia Conference 96. Boston, MA*, pp. 18–22, 1996.

[KB98] F. Kensing and J. Blomberg. Participatory design: Issues and concerns. In *Computer Supported Cooperative Work*, Vol. 7, pp. 167–185, 1998.

[KSB98] F. Kensing, J. Simonsen, and S. Bodker. Participatory design at a radio station. In *Computer Supported Cooperative Work*, Vol. 7, pp. 243–271, 1998.

[L.02] L. レッシグ, 山形浩生訳. コモンズ. 翔泳社, 2002.

[Les00] L. Lessig. *Code and Other Laws of Cyberspace.* Basic Books, 2000.

[Les01] L. Lessig. *The Future of Ideas: The Fate of the Commons in a Connected World.* Random House, 2001.

[LW91] J. Lave and E. Wenger. *Situated Learning: Legitimate peripheral participation.* Cambridge: Cambridge University Press, 1991.

[New98] S. Newman. Here, there, and nowhere at all: Distribution, negotiation, and virtuality in postmodern ethnography and engineering. In *Knowledge and Society*, Vol. 11, pp. 235–267, 1998.

[NLHL03] Mark W. Newman, James Lin, Jason I. Hong, and James A. Landay. Denim: An informal web site design tool inspired by observations of practice. In *Human-Computer Interaction*, Vol. 18, pp. 259–324, 2003.

[NOT] NOTA.
http://www.nota.jp/.

[NPO パ] NPO パンゲア.
http://www.pangaean.org/.

[Rov87] Winston W. Rovce. Managing the development of large software systems: concepts and techniques. In *Proceedings of the 9th international conference on Software Engineering*, pp. 328 – 338, 1987.

[Sta02] R. M. Stallman. *Free Software, Free Society: Selected Essays of Richard M. Stallman.* Free Software Foundation, 2002.

[Wen90] E. Wenger. *Toward a theory of culture transparency -Elements of a social discourse of the visible and the invisible-.* PhD thesis, Irvine: University of California Irvine, 1990.

[Wen98] E. Wenger. *Communities of practice.* Cambridge: Cambridge University Press, 1998.

[永田] 永田周一. Nota api マニュアル.
http://nota.jp/plugin/.

[永田 04a] 永田周一. Nota - web の新しいコミュニケーションツール. 第 12 回インタラクティブシステムとソフトウェアに関するワークショップ（WISS 2004）, 2004.

[永田 04b] 永田周一. なめらかなファイル操作体系の構築と実用化. 平成 15 年度未踏ソフトウェア創造事業成果報告書, 2004.

[高崎 06] 高崎俊之, 永田周一, 安川直樹, 石田亨. 共生プログラミングが情報社会を救う. 2006 年「情報社会のデザイン」シンポジウム（第 1 回）, 2006. http://www.ai.soc.i.kyoto-u.ac.jp/infosociety2006/panel_discussion1130.html.

[上野 06a] 上野直樹. ネットワークとしての状況論. 文化と状況的学習—実践、言語、人工物へのアクセスのデザイン. 凡人社, 2006.

[上野 06b] 上野直樹, ソーヤーりえこ, 永田周一. 学習環境のデザインのためのネットワーク指向アプローチ. 科学技術実践のフィールドワーク, pp. 56–74. せりか書房, 2006.

[神原 06] 神原啓介. Web 上での協同イラストレーションシステムの制作. 慶應義塾大学 修士論文, 2006.

[増井 03] 増井俊之. インターネット上の新しいコミュニケーションツール. 情報処理学会ヒュー

マンインタフェース研究会研究報告 (招待講演), 2003.

[田川] 田川欣哉. レーザードローイングツール「afterglow」. http://www.lleedd.com/afterglow/.

[梅田 06] 梅田望夫. ウェブ進化論. 筑摩書房, 2006.

[鈴木 06] 鈴木健. Adoc と呼ばれる新潮流. 情報通信ジャーナル, 第 24 巻, 2006.

付録 A

NOTA ネットワークの活動履歴

2005/07/31 武蔵工大で NOTA ワークショップ開催
2005/10/30 NOTA ネットワークメーリングリスト立ち上げ
2005/11/19 プラグイン・プロジェクト ミーティングを開催
2005/12/18 nota@bar を開催
2006/01/09 NOTA ネットワーク関西 キックオフミーティングを開催
2006/01/23 NOTA プラグイン仕様書を公開
2006/02/17 NOTA ネットワーク横浜ミーティング
2006/06/01 トライアル版に 8 つのテーマの NOTA を設置
2006/06/01 NOTA サイトをリニューアル
2006/07/24 トライアル版のデザインを改良
2006/07/30 ワークショップ「ウェブとコミュニティ形成」を開催
2006/09/08 市民メディアサミット 06 の「市民メディアにおける新しいネットツールの活用」というセッションで NOTA を発表
2006/09/08 NOTA ダウンロード版を一般公開
2006/09/13 横浜市市民活動支援センターの IT 講習会にて、NOTA の講習会を行う
2006/09/26 クリエイティブ・コモンズセミナーで NOTA および NOTA を利用した「C-Shirts」プロジェクトを発表
2006/10/26 REMIX TOKYO で NOTA を発表

付録 B

NOTA ネットワーク運営方針

NOTA のプロジェクトは、「この指止まれ方式」で進んでいます。

この指止まれ方式というのは、何かをやりたい人がメーリングリストで提案し、賛同者が集まればその人を中心に行動するというやり方のことです。

具体的には、以下の 3 つにまとめられます。

1. フィードバックを尊重しています
   提案や投稿に対して、「よいアイデアだね」というように積極的に意見や感想を返信（フィードバック）しましょう。フィードバックは、やる気のみなもとです。
2. 全員で共有します
   考えていることは必ずメーリングリストで提案します。そうすることで、一部の人だけでプロジェクトが進まないように気をつけます。逆にいうと、提案すれば誰でも好きなプロジェクトを立ち上げて、進めることができます！
3. 外に対してオープンに
   ここで出てきた成果は、どんどん外に出していきましょう。NOTA ネットワークで得た体験や知識をメンバーがそれぞれ持ち帰って、様々な媒体で伝えていきましょう。

作成日：2005/10/29

NOTA：ユーザー主導型コミュニティ活動支援システムの研究

2007年 3月30日　　初版発行

著者　永田周一

監修　安村通晃

発行　慶應義塾大学 湘南藤沢学会
〒252-0816　神奈川県藤沢市遠藤5322
TEL：0466-49-3437

Printed in Japan　　印刷・製本　　ワキプリントピア

ISBN 978－4－87762－174－2
SFC-MT 2006-004

■本論文は修士論文において優秀と認められ、出版されたものです。