

# ギークのコミュニティ： Web を媒介としたプログラミング学習環境

松村 飛志

近年 Web テクノロジーが社会的に発達した事によって、従来のプログラミングの状況も大きく変化している。こうした発展の影には、歴史的に Web で活動する先駆的なユーザー＝ギークのコミュニティが重要な役割を占めていた。ギークの重要性は近年エンジニアリングやマーケティングの観点では認められつつあったが、本研究では特にギークの学習環境に着目し、生活や活動に根付いたプログラミング技術習得の過程や文化的実践としてのプログラミングなどを調査・記述した。また、そうしたギークのコミュニティをフィールドワークするなかで、ギークたちがテクノロジーとどのような態度で付き合っているのかということや、ギークの情報エコロジーが一個人の作ったサービスによって一変するような、柔軟で複雑な再デザインの構造であることを明らかにした。さらに今後もこうしたギークのコミュニティへアクセスすることによってテクノロジー開発や運用の新たな可能性が拓かれることを示唆する。

キーワード：ギーク、ハイブリッド・コミュニティ、学習環境のデザイン、情報エコロジー、エスノグラフィー

## 1 はじめに

近年、Web2.0 というキーワードに代表されるように、特に Web を中心とした ICT は社会的に大きな影響力を持ちつつある。そうした変化の中において、プログラミングという行為そのものが変化するのもまた必然的なことである。Web を媒介とした現代のプログラミング実践は、従来のプログラミング学習とは違ったプロセスとなっているのである。

そこで本論文では、Web を媒介とした現代的なプログラミングの学習環境を記述することを目的とする。また、そうした学習環境の記述を通じて Web テクノロジーの技術社会的な布置の変遷やエンジニア、ギークたちの情報エコロジーを調査分析することも可能であろう。

Web エンジニアたちのコミュニティは、テクノロジーとヒトが複雑に混在した場である。これは従来より学習理論において主要なトピックとなりつつあった“ハイブリッドなコミュニティ”のダイナミックな実例でもある。こうしたコミュニティをエスノグラフィーによって調査することによって、技術と社会の関係を探っていくことも状況的学習論の一部として重要である。

現代におけるプログラミング学習は、Web という情報環境と切り離せないものとなってきている。本研究では

こうした近年のプログラミングをめぐるめざましい状況の変化を自身の経験を元に記述し、さらに Web テクノロジーをめぐる技術社会的な布置の変遷やエンジニアたちの情報エコロジーを調査分析することで、現代的なプログラミング環境を明らかにし、プログラミング教育のための新しい学習環境のあり方を探る。

また、工学的技術と社会の関係については、従来から様々な意見があった。

たとえば情報社会への考察などで知られる哲学者・東浩紀は、近年の著作においては工学的な技術が社会の環境管理に導入されるという、テクノロジーと権力に対する悲観的なビジョンがあった。

しかし筆者は実際にテクノロジーと共に生きているギークたちの活動をフィールドワークによって調査した結果、まったく別のテクノロジーの姿があるのではないかと考えている。

そのような社会の工学的不安をめぐる諸言説に対しても、ギークたちの実践をより詳細に調査していくことで、新たな提言が可能となるのではないだろうか。

### 1.1 Web とギーク

本論文で主役となるのは Web2.0 の開発主体である Web 系エンジニアや、Web2.0 のコアユーザーであるギークと呼ばれる人々である。まずはこのギークについてこれまでの議論を紹介しておこう。

広い意味でのコンピュータオタクを意味する単語には、hacker, geek, nerd など様々なものが挙げられる。しか

しそのなかでも、Web において近年時に親しみのある呼び名は geek だと思われる。これらギークたちの情報行動を知ることが、最新の Web テクノロジーはもとより、Web を中心とした現代的なプログラミングのスタイルを探ることに繋がっている。

こうしたギークたちは「新しいテクノロジー」を積極的に取り入れるという志向性が最大の特徴であり、新開発された商品の普及過程においてもオピニオンリーダーあるいはアーリーアダプターといった、初期の重要な役割を担っているのである。

近年では電子機器メーカーや Web 系ベンチャー企業においても、こうしたサービスの最初期にコアなユーザー層を形成するギークのコミュニティが、Web テクノロジーを中心としたマーケットを発展させるうえで先進的な役割を担っているという認識は定着しつつあると言っていいたろう。

特に Web においては大企業が作るサービスも個人が作るサービスも、最初に使うのはこうしたギークのコミュニティであるため、こうしたユーザーたちとの良好な関係を構築すること商品の普及過程において重要と考えられているのである。

また、近年はギークのなかでも Web というメディアの中で注目を集め、特に先進的なリーダー的立場を意味する言葉としてアルファブロッガーやアルファギークという概念も生み出されてきた。アルファギークとは、「産業を変化させる力を持つ新しい技術に早いうちに飛びつき、ああでもないこうでもないといじくっているうちに、技術が進むべき方向性を示し始める、先鋭的で飽きっぽいエンジニア」のように定義付けられている。こうしたアルファギークたちはギークコミュニティの中心的人物であり、彼ら自身も新たな Web サービスの開発者として活躍している事が多い。

上記のように、ギークのコミュニティはエンジニアリングの観点やマーケティングの観点ではようやく評価され始めたが、こうした分野だけではなく状況的学習論の観点・コンピューター教育の観点からも彼らのスタイルを学んで行く必要があるのではないだろうか。

後に詳しく記述していくが、ギークたちは自分の欲望や生活に直結した形で、非常に実用的に・苦を厭わずにコンピューターテクノロジーを駆使している。これは企業に属して仕事としてプログラミングを行うという従来のプログラマー観が古いものと成りつつあることを意味している。すなわち、現代的な Web の情報生態系においては職業的エンジニア/プログラマーだけでなく、一般の Web ユーザーへの開かれた文化的な実践としてプログラミングが捉えなおされつつあるということである。こうした状況は、今後もより安価に普及するであろう ICT によってより多くの人にプログラミングを道具として使い

こなす可能性が開かれていること示唆している。

次章では、こうしたギークたちの情報エコロジーについて記述していく。

## 2 ギークの情報エコロジーの変遷

本章では、2000 年以降の Web におけるギークたちの情報エコロジーの変遷についてまとめていく。

ギークたちは、Web を中心にコミュニケーション、エンターテインメント体験や情報収集・発信活動を行っているため、タブ表示のブラウザを愛用している場合がほとんどである。通常インストールされている Internet Explorer では一つのウィンドウに 1 サイトしか表示できず、大量の Web サイトを閲覧するという目的を達成することが非常に難しいからである。

タブ型ブラウザは Web2.0 以前から膨大な数が開発されており、開発競争はブラウザ戦争とも呼ばれている。現在の Web の状況を変えた大きな転機として 2002 年頃からオープンソースのタブ型 Web ブラウザである Firefox が開発されはじめ、さらに 2005 年にはそれまで有償ライセンス商品であった Opera が無料で提供されはじめた。これらのブラウザが従来にはないモダンな機能を備えていたことでギークたちの Web 閲覧方法もより高度で複雑なものへとなっていった。

Web2.0 初期においては、Google 検索や bloglines、del.icio.us はてなブックマークなどがギークたちの情報行動の中心的なツールだった。

Web2.0 における情報エコロジーのもっとも大きな転換点として RSS と呼ばれる更新通知規格の展開が挙げられる。ギークたちは大量の Web サイトを巡回対象としてきたが、それらを機械により効率的に行わせるためのアイデアが RSS だった。しかし RSS は単なる更新通知規格としてだけでなく、さまざまなサイトの機能を相互に融合させる「マッシュアップ」という新たなプログラミングのスタイルをも生み出すことになっていく。

Bloglines や LivedoorReader のような RSS リーダーと呼ばれる Web サービスの登場によってギークたちは Web 上に増え続ける大量の情報源をチェックするという作業を自動化させてきたのである。

Web の更新を自動検出して通知するしくみは Web2.0 以前から WWW などのアプリケーションによって実現されてきたが、RSS はそうした更新チェックをよりインテリジェントに行うために考え出された規格である。WWW はサイトのヘッダ情報などから更新をチェックしていたが、RSS は Web サイトのどこが変更されたのかという情報を具体的に含んでいるのが特徴である。

Bloglines や LivedoorReader など ASP 型の RSS リーダーの特徴は、Web 上にあるためどのパソコンからでも口

グインすれば使える 更新された情報の未読既読管理(どれを読んで、どれを読まなかったか)が可能である、サーバー側のクローラーが各サイトの更新チェックを一括で行うのでインターネット全体の負荷が抑えられるなどである。

また、Web2.0 においては様々な情報共有サービスが Web 上に構築されていった。日記形式で相互に情報発信する Blog が最初期の代表的なものであるが、写真共有サービスの Flickr や動画共有サービスの Youtube は現在でもギークたちに愛用されているサービスである。

Web2.0 において成功したサービスは、特定の情報に対して十分な機能を備えた Web サービスであった。また、これらのサービスが API などを通じて情報を公開しているというのも特筆すべき点である。これらの API を利用することによって各サイトにアップロードされたデータは自由に再利用が可能な状態でホスティングされる。このデータの自由な再利用が、ギークたちにとっては重要なことであった。また、Last.fm のように、再生した音楽を自動記録し、その情報を元に音楽をオススメするというシンプルなサービスなども根強いユーザーを有している。

ギークたちにとってこれらのサービスはそれぞれが自己表現やコミュニケーションのためのチャンネルとして機能している。現在の Web ではこうした細かい単位に特化した情報発信はさらに細分化されていると考えていいだろう。このように Web 上に自分の生活情報を豊富に載せることが出来るようになったため、他者の情報行動を用意に覗き見ることが出来るようになってきているという事が重要である。

また、2006 年に始まった Twitter というサービスは、ギークの情報エコロジーを大きく再編した重要なサービスである。これはマイクロブログと呼ばれる新しいジャンルのサービスで、Web 上にチャットのように「ひとりごと」を時系列で次々と発信することができるようなサービスである。Twitter はこれまでのチャットサービスや掲示板のように「特定のテーマについての場」に個人が集まる方式とは違い、あくまで「個人のひとりごとがそれぞれのページにオープンな状態で載っている」というサービスである。この Twitter によって、ギークたちはこれまでの SNS とは文字通り一桁違う数の人間と極めて容易に関わることが出来るようになった。Twitter では、数百人～数千人のユーザーの発言を「Follow」(mixi でいう「マイミクに追加」)し、発言を受信するのが普通である。従来の mixi や GREE のようなリアルでの知人と関係する事を前提として緊密かつ深いコミュニケーションを志向していた SNS とは違い、Twitter は圧倒的に浅く広く・ちょっと興味を惹かれた他者とも気軽に関わることが可能なようにシステムがデザインされているので

ある。

この Twitter は Blog や SNS に比べても遥かにシンプルなサービスだが、API などが豊富に用意されており外部から自由に情報の出し入れが可能なものだったため、様々な情報を混ぜ合わせるためのプラットフォームとしてデファクト・スタンダードとなりつつある。API を応用したサービスとして、movatwitter、Twitter 検索や Favotter のような日本人ユーザーによるマッシュアップサイトも生まれた。これらは API を利用して本来 Twitter になかったサービスを独自に実現したもので、movatwitter は日本のケータイから Twitter へ一言を投稿できるようにし、Twitter 検索は文字通り Twitter 全体から特定の単語で検索することを、そして Favotter は Twitter の発言ひとつひとつに Favorites という目印が付けられる機能を可視化するためのサービスである。

こうした一個人や素人によるマッシュアップサイトが、ギークの Web 情報エコロジー全体に対して決定的な影響力を持ちうるということが、ポスト Web2.0 においては極めて重要な状況だと考えられる。

Twitter そのものは API によって特定の形式でデータを出し入れできるプラットフォームであり、大雑把な使い方は提案するが再デザインの余地が無限に残されている。

本来 PC 向けのサービスで日本のモバイル機器では使えなかった Twitter だが、movatwitter によってケータイ向けインターフェイスが用意されたことで身近なできごとを写真付きでアップロードするというライフログ的な使い方も可能になった。

さらに Twitter 検索によって、ギークは自分たちが共有している俗語・ネットスラングを検索することによって、特定の話題に言及しているユーザーを簡単に見つけ出し、新たな話題を見つけたりそのユーザーを Follow して関係を持つといったことも可能になった。

また、Favotter は自分の発言のなかで何が注目されているのかという事や、Twitter に発信される膨大な発言のなかで何が特に注目されているのかという事を可視化することが出来るようになった。

ポスト Web2.0 で起こりうることの一つに、こうしたライフログのさらなる細分化が挙げられる。従来ライフログの研究ではコピキタスコンピューティングの技術などを半ば無理やり現実の生活の中に取り入れ、物理的行動あるいは社会的コミュニケーションなどまでを含めてすべてデジタル化し保存しようという研究が主流であったが、現在 Web で流行しているこれらのライフログサービスは、コミュニケーションやエンターテイメント体験、情報収集・発信など情報活動の大半を Web 上で行っているギークたちの行動をそのまま抽出して一箇所に集めることで様々な付加価値を生み出すという方向性へ進みつ

つある。もともとデジタルなこれらの情報行動のログは、無理なく収集することができるうえ、Web2.0の典型的概念である集合知のアイデアによってソーシャルな価値を生み出すという考え方は受け入れやすいものと言えるだろう。

つい最近では Last.fm のようにニコニコ動画や Youtube の再生履歴を自動で記録して公開できる「mitter」というサービスや、ブラウザの Web サイト閲覧履歴をすべて自動記録して Web の視聴率を統計的に調査する「Pathraq」などがリリースされており、特定の分野に特化した個人の情報行動をログに取り、集合知の実現を目指すサービスは今後もどんどん増えていくものと思われる。

さて、こうして個人の多様な情報が Web のなかに秩序無きまま散在しつつある現在の Web だが、一方で Web に分散して溶け出す個人情報を「まとめる」ためのサービスや仕組みも作られてきた。こうした役割を担ってきたのは、主に Web プロフィール生成サービスである。2007 年になって相次いで iddy, tako3.com, fooo.name!! などのサービスがリリースされた。それぞれ独自の特色あるプロフィールサービスだが、これらのサービスに共通しているのは「複数のサービスに登録しているユーザーの情報をまとめる」という機能を持っているという点である。iddy は自分の情報を自分でまとめて公開する自己紹介サービスだが、tako3.com や fooo.name は wiki のように自分の情報について他人も編集可能なサービスであることは特筆すべき点である。

こうしたサービスの登場によって、これまでなら Google などを使いながら気になるユーザーの利用しているサービスを探してチェックしていく必要があったところが、twitter で新しいユーザーを発掘して iddy, tako3.com, fooo.name!! などをそれぞれ参照し、そこから個人の Blog やはてなブックマーク Flickr や Youtube, Last.fm などこれまでの定番となっている Web2.0 サービスへと辿っていく新たな情報行動の流れが出来上がった。さらに 2008 年になって Google も Social Graph API と呼ばれるサービスをリリースした。

このような誰が何を使っているのかを一望することができるサービスの登場による影響は、ただ効率化というだけではない。こうしたサービスを利用することによって、ギークはお互いの興味関心やパーソナリティを自然と知ることになっていく。そして、こうした情報エコロジーは、単なる技術の習得を超えた出会いへと繋がっているのである。

近年 Twitter やその同時期の情報サービスを契機とした情報生態系の再編によって、ギークたちの情報行動は大きな影響を受けつつある。リアルでの生活が Web へ流れ、さらに Web 上で形成された関係がリアルへと流れ Web

とリアルを隔てていた境界線が希薄になっていき、Web・リアル両方での接触が加速していく。

次章では、こうした Web に対するリアルでのギークたちの活動へのフィールドワークについて記述していく。

### 3 リアルにおける活動

ここでいう「リアル」とは、ヴァーチャルな空間である Web と対比したスラング的意味合いを持っているが、本章では、そうしたギークとしての「リアル」におけるフィールドワークについて記述していく。具体的にはさまざまなコミュニティのイベントにメンバーとして参与することで、彼らがどのような価値観や行動規範を持っているのか、そしてどのように様々なテクノロジーと付き合っているのかということに焦点を当てて調査した。

エンジニアたちは、上で述べた mixi や twitter, blog などを通じてお互いに知り合い、オフ会や特定のテーマに沿った技術の勉強会、開発合宿、カンファレンスなどを毎週と言っていいほど非常に頻繁に行なっている。こうした活動は各自が所属している学校・企業など組織の枠に捕らわれておらず、オープンで多様な参加者を許容するものである。

私がこれまで参加したものには、Twitter ユーザーが渋谷で集まった Twitter オフ会（参加者 50 名）や Firefox などの Mozilla プロダクトに関するカンファレンスである Mozilla24、そして昨年末に行われた Twitter 忘年会（参加者 150 名）などである。

こうした勉強会はリアルで参加できる人たちだけが独占するのではなく、ustream という Flash 映像配信サービスのおかげで、参加できなくても閲覧して、チャット経由で突っ込みを入れたりすることができる環境が整い始めている。現在ではほとんどの勉強会などでこうした試みが成されており、さらにライブ中継だけでなく録画もしておいて、ニコニコ動画にアップロードするといったことも頻繁に行われている。

ustream に限らず、テクノロジーを自分たちの実践を



図 1 ネットで生中継される勉強会

オープンにしていくために活用していく志向性があると言える。

### 3.1 秋葉原でのギークたちの実践

秋葉原はやはりギークたちにとって特別な町である。先述した Twitter 検索などで秋葉原での活動を観察してみると、毎日のように誰かが秋葉原に行って買い物をしているのがわかる。さらに毎週末には小規模なオフ会や飲み会が誰が言い出すわけでもなくごく自然に催されている。



図2 路上の活動を生中継するシステム

秋葉原という都市でのギークたちのテクノロジー的実践で特筆すべき点として、彼らは市販のハードウェアと複数のネットサービスを組み合わせることによって自分たちの活動を移動しながらストリーム配信して、ネットのユーザーたちと共有することを可能にしているという点が挙げられる。

前節で述べた勉強会のような公的なイベントだけでなく、自分たちの都市でのプライベートな日常、遊びなども ustream によって配信しているのである。

こうした日常の配信は場合によっては100人以上が視聴することもあり、あたかも映画「トゥルーマンショー」のような状況となっている。メディア論的にも、都市論的にもまったくあたらしい実践が現実に行われているのである。

このように、技術的リテラシーが非常に高いユーザーたちにとっては、すでにユビキタスネットワークは日常の実践の一部となっているといっていいたいだろう。

ここで重要なのは、彼らギークたちの技術的リテラシーを高めているのは、さまざまなサービス/ソフト/ハードを含む製品の使い方を熟知し、カスタマイズし、自分の目的に応じて組み合わせて使いこなすといった道具と知識の布置連関であるということである。

また、そうした布置連関によって得られた技術的リテ

ラシーが、彼らの主体性・価値観や都市での具体的実践といったものに非常に大きな影響を及ぼしていると思われる。

こうした実践が生まれたのは、秋葉原というテクノロジーを中心としたモノとヒトが溢れている都市でこそのものであると言えるだろう。



図3 秋葉原にある Linux カフェ

秋葉原の裏通りに、Linux カフェという喫茶店がある。一見すると普通の喫茶店だが、実はここは「リナカフェ」と呼ばれ、さながらギークたちの秘密基地のように頻繁に待ち合わせ場所として利用される、秋葉原の中でも特に重要なスポットとなっている。

カフェは、前世紀西洋では見知らぬ隣人たちとコミュニケーションを行う場であり、新聞記者などが時事情報を集めるなどメディア論的にも重要な場であったが、近代日本ではカフェのそのような機能は薄れ、親しい友人との休憩のための場という側面が強くなっていた。

しかし秋葉原の Linux カフェは Web とハイブリットなコミュニケーションによって、初対面のネットユーザー同士がコミュニケーションを行うという、ギークたちのねじれた公共圏として成立しているのである。Linux カフェがこのような見知らぬギークたちの集会所となった過程には様々な要因が介在していると考えられる。第一に、店内の電源が開放されており自由に使えるということや、インターネットにアクセス可能な wifi ホットスポットなどがあるということなどである。こうした電源や電波といったサービスはギークたちにとっては水よりも重要なリソースなのである。

また、オフ会などで盛り上がってしまって騒いでもあまり怒られないというのも重要な要素のひとつと言えるだろう。

## 4 インフォーマルなプログラミング実践

続いて、現代的プログラマーたちがどのように文化的実践としてのプログラミングを行っているのかということについて、私が見たことを述べていこう。

### 4.1 生活を改善するためのプログラミング

Web においては、一人ひとりが自由に使うことのできる Linux サーバーがあれば、Web の情報を自動で収集する基地のようなものが構築できる。そのため、ほとんどのギークはこうした自由にプログラミング可能な情報収集発信基地を作り上げている。たとえ自分がオフラインでも、寝ている間であってもこうした基地は作動しつづけ、決められた手段で情報を収集・処理し続けているのである。

彼らがそうした情報基地で行っているのは、たとえば「ニコニコ動画の最新の人気動画を自動でダウンロードし iPod に転送する」であるとか、「Web にある膨大なイラストなどの画像ファイルをジャンル・キーワードとペアで検索できるようにデータベースを構築する」であるとか、「さまざまな匿名掲示板のログを自動でインデックス化して全文検索できるようにデータベースを構築する」などといった、自分たちの日常的な Web での生活や遊びに直結したことである。これは先述した各企業・個人が公開している API などとオープンソースのソフトウェアなどをうまく組み合わせて手軽に作り上げられている場合がほとんどであり、すべて一からソフトウェアを書いているとは限らない。

現代的プログラマーがこうした実践を熱心に行う背景には、ギーク独特の価値観や行動規範が重要な役割を果たしているものと思われる。具体的には、機械にできることは人間は可能な限りやらないという「自動化」や、手間の掛かる作業や繰り返しの作業、面倒なことは可能な限り避けるという「怠惰」などである。

ギークたちにとってはプログラミングは面倒な課題や仕事などではなく、うまくプログラミングすることによってどんどん自分の時間を作り出すことができるような情報生活を改善するための手段なのである。

近年ではさらに、ちょっと勉強しただけでも Greasemonkey のようなツールによって、Web サイトのインターフェイスを直接改善することも可能となってきた。たとえば、Greasemonkey で実現されているのは Google と Yahoo と Wikipedia と一括で検索することや、Google の検索結果にはてなブックマークの登録数を表示するといった事である。

こうしたことは結局、技術的なリテラシーとはテクノロジーを「与えられたまま」使うのではなく、ツールを駆使してそれらを再度分解し、自分にとって使いやすい

ように布置連関を再編する行為を通じてしか得られない、とまとめることができるのではないだろうか。

以下では、それぞれの実践を以下で詳しく見ていこう。

### 4.1.1 Greasemonkey

機能を拡張するための仕組みはアドオン (Addon) と呼ばれている。さまざまなアドオンを組み替えていくことで、Firefox はそれぞれまったく違うブラウザのような機能を持ちうるのである。

そうしたアドオンのなかでも特にギークたちにとって重要なツールとなっているのが、Greasemonkey と呼ばれるアドオンである。

Greasemonkey は `user.js` と呼ばれるファイルを読み込ませることで、Javascript によって特定の Web サイトの利用方法を改善したり根本から書き換えてしまうことができるアドオンである。

Web サイトは従来、開発者が独断で作ったただひとつのサイトをそのまま利用するだけの場合がほとんどであったが、Greasemonkey の登場によって、Javascript を理解できる Web ユーザーはサイト開発者が気付かずに改善してこなかったような部分を勝手に改善してしまうことが可能になり、さらにスクリプトを配布してその勝手な改善を他のユーザーとも共有することが可能になったのである。

これは Web サイト製作というものの価値観を根底から変えなければならないほど重大な変化と言っていいだろう。従来までは、開発者は「ユーザーたちの使いやすさ」を一方向的に想定して Web サイトを作っていたが、これからはヘビーユーザーたちがなるべく改善しやすいような設計にすることを心がけることが、結局最もユーザビリティを高めることに繋がる場合もあるということの意味している。たとえば全面 Flash のサイトがどれほど作りこまれていても一部のギークたちにあまり好まれないのは、Greasemonkey によって好き勝手に書き換えることが技術的に若干難しいという理由であったりもする。

### 4.2 CodeRepos をはじめとするバージョン管理システム

前述した様々なスクリプトファイルを公開したり、他人と一緒に開発作業を進めようと思った場合、さまざまな技術・社会的問題が発生する。たとえば同時刻に上書きが発生して混乱したり、誰がどこをいつ書き換えたのかわからなくなってしまふなどである。

こうした問題を解消するために、従来から様々なバージョン管理システムが利用されてきた。近年では `subversion` というオープンソースのバージョン管理システムが流行している。このシステムが好んで利用されている理由として、`trac` というプロジェクト管理・バグ

管理システムと手軽に連携することが可能という点がある。オープンソースのプロジェクトでは様々な小さななければいけない仕事 = タスクが発生し、それを参加メンバーたちの間で割り振ったりしながら進めていくのが定番である。trac はこうした作業の割り振りを行うためのシステムだ。subversion と trac を連携させることによって手軽に仲間たちとオープンソースのプロジェクトを立ち上げ、協同的にプログラミングを行う環境が構築可能となった。

逆説的に、オープンソースのプロジェクトへ参加したりまとまったソースコードを閲覧するための最大の近道であり最低限の共通の手段は、subversion ということになる。他者と協同的に活動する現代的プログラミングの実践には、こうした subversion や trac の作法をお互いが理解しているということが欠かせない。たいていの Web 系企業は社内でこうしたバージョン管理システムを用いて、プログラムを社員がだれでも閲覧・変更可能にしているものと思われる。

さらに、オープンソースや企業などフォーマルなプログラミングのプロジェクトの場面以外でも、こうしたバージョン管理システムを利用して情報を公開しようという実践が行われている。公開リポジトリや個人リポジトリと呼ばれているそれらのプログラム置き場は、前述した Greasemonkey の user.js や、vim の設定を記述したスクリプトファイル、あるいはちょっとした実験的プログラムなどを手軽に公開するための手段として定着してきた。それらのプログラム置き場はたいていの場合一人で使うために設置されており、整理されておらず、趣味でプログラミングをやっている無名プログラマの場合は注目されているとも限らなかった。

そうした分散していた個人の細かいプログラム置き場をひとつの場所に集約してみようというのが poderepos と名づけられたプロジェクトのレポジトリである。

coderepos は現在200名以上のプログラマ/非職業プログラマが参加しているが、全員で力を合わせて巨大なプログラムを作っているわけではない。参加メンバーたちが小さい便利なプログラムを持ち寄ったり、設定ファイルを公開するための置き場所として利用している。coderepos に置かれているファイルはあらゆるプログラミング言語におよび、閲覧やダウンロードなどの利用は参加しなくとも誰でも行うことができる。

coderepos のトップページには以下のように記述されている

「みんながそれぞれ作って公開してる公開レポジトリを一緒にたにしちゃいたい。参加してる全員がどのファイルもみたり変更したりできるような。

パッチ送られてくる代わりに「後で見とくからコミットしといて」とかいいたりとか、つくりかけで放置した



図4 Coderepos プロジェクト

もので他の人が興味もったら続き作ってもらおうとか、メンテするのめんどくなったのだからかやしてもらおうとか、突発的に誰かと一緒にプロジェクト始めたりとか、できる！」

この文言が、まさに現代的プログラマーたちの生態、プログラミングのスタイルをそのまま言い表したものであると言ってもいいだろう。

## 5 Web プログラミング技術習得の学習プロセス：ケーススタディ

これまで述べてきたような学習・開発環境のなかでギークは活動を行っている。ここでは、私がフィールドワークの過程で知り合ったギークの具体的な事例について記述する。

私が Twitter で知り合った小野マトベ氏は、2005年に Web2.0 のブームに触れてはじめて Web プログラミングの勉強を始めたギークの一人である。大学では情報学部の学生ではあるが専攻は CG・画像処理などであり、大学の講義では Web テクノロジーについて学ぶ機会はほとんど無かったという。そうした状況でありながら、Web2.0 初期の時点では技術書や Google 検索などを駆使することによって Web サービスを一通り個人で作れるようになったという。そして彼は Amazon の書評などの商品レビューを自動で取得し、新しい商品と出会うことができるサイト「レビューライン」を開発した。当然、自分が作ったサービスを公開するためには Web プログラミングの技術だけではなくドメインを取得したり、レンタルサーバーを契約したりする必要があるがそうしたことも書籍や Google 検索などで容易に得ることができるという。また、完成したサイトは blog サービス「はてなダイアリー」やソーシャルブックマーク「はてなブックマーク」に宣伝し、ギークたちの注目を引くことができたという。

さらに先に紹介した、2006年にはじまった Twitter の発言から人気のあるものを抽出するサービス Favotter を開発したのも小野マトベ氏である。Favotter が公開さ



れたのは2007年11月であり、つい先日のことだ。小野マトベ氏によれば、Twitter の登場によってサービス開発や普及の過程も大きく変わったという。従来であれば書籍や検索でたどり着くしかなかった技術に関する情報が、Twitter でプログラミングの疑問点やサーバーのチューニング方法などが人づてに情報が得られるようになり、また、サイトが完成したあとの告知やサービス改善のための運用上のフィードバックなども、Twitter の口コミなどによって広まっていたという。こうした一連の変化はもちろんFavotterがTwitterのマッシュアップサービスであることによってスムーズに可能となった面もあるが、多くのギークがTwitterに参加しつつある現状から、今後もTwitterをベースとしたWebサービスの開発・運用というスタイルは定着するものと思われる。

また、本学でも上野研究室の筆者・松村や土橋研究室瀧本がWebサービスを開発・運用していたが、やはりGoogle検索などによって必要な技術を探し、また、参考になるソースコードのサンプルやオープンソースのライブラリなどを見つけるという過程が重要なものだった。筆者の場合だと、事例研のために作ったグラフィティのサイトをとりあえず公開しておいたところ、NPO 法人の方から声を掛けていただけるなどの過程があった。できあがったサイトをどのように告知し、運用していくのかということも開発と同じくらい重要なことである。

### 5.1 アクセスの可能性

さらにWebとリアル両方からのアプローチによって、以下のようなことが明らかとなった。

プログラミング習得においては「次のステップ」が明らかになっていないということが問題となりがちである。プログラミングでは具体的な道具や環境が目に見えるものではない。基本的にはエンジニアは自身のパソコンを中心に開発環境を構築しているが、パソコンはパーソナルな環境であり、プライベートな情報を多く含むため、ありのままを公開することが難しい。こうしたワークスペースへのアクセスが制限されているため、自分よりひとつ上のステップに位置するヒトのノウハウなどが見えにくいという問題がある。

こうした問題を補うのが、web2.0的なテクノロジーを媒介にした現代的なプログラマーのコミュニティである。具体的には、RSSリーダーによる膨大な情報のチェック、ソーシャルブックマークによる技術トレンドの共有、Wikiによるまとめサイトやblogにおけるちょっとしたスクリプトの公開、Subversionによる細かな設定ファイルの共有などが挙げられる。

また、現代的プログラマーの世界では、特定の技術への関心を中心としてコミュニティができあがっており、そうしたコミュニティへの参加によって技術をより効率

的に習得することが可能となる。こうしたコミュニティは、TwitterなどのWebサイト上のヴァーチャルなコミュニティ及びリアルなつきあいを含んだ、ハイブリッドなものである。

技術や情報は単なる抽象的な概念ではなく、ヒトを媒介として広がっていく。単に技術単体だけを見るのではなく、面白い事をやっているヒト、その分野で有名となっているヒト、など他者への興味関心をベースにこうしたコミュニティを見ていくことによって、キーとなるトピックを常に追うことが可能となり、さらにそのヒトが興味を持っている他の技術を発見することも可能となってくる。

さらに、従来の教育ではプログラミングを学習していく上での個人的なモチベーションを見出すのが難しいという問題もあげられるが、今回のフィールドワークによって、ギークたちにとってはプログラミングを中心としたさまざまなテクノロジーは常に特別な仕事の為の道具であるとはかぎらず、遊びや生活、そして世界をより豊かにしていくことに活用できるようなものであるということが明らかになった。そうした仕事や課題を超えたインフォーマルなプログラミング実践によって、単なる「技術の習得」を超えた「技術の身体化」へと繋がるのである。

### 5.2 歪んだ情報エコロジーのデザイン

以上で見えてきたように、現代のWebを媒介としたギークのコミュニティは様々なヒト・テクノロジーが入り乱れ、日々刻々と再編される場であることがわかる。

大きな流れをまとめておくと、90年代のホームページの時代にはYahoo!のような人力でカテゴリ分類を行うポータル検索サイトがWebでの情報行動の中心的なスタート地点であったが、その後すぐにGoogle検索やBlog、SBSなどが中心の個が分散したりズーム的なコミュニケーションへ、そしてTwitterやSocial Graphによる再統合個人を中心としたコミュニケーションへと至りつつある。

このように、Web全体の生態系やギークの情報行動は海外のまったく新しいサービスによって大枠が再編されてきた。しかし近年のAPI公開やマッシュアップなどの状況によって、デザインされる側であるギーク自身が、自らツールを生み出すことによっても場を再デザインすることを可能となっているのである。開発者もそれを最初に使うギークも、基本的には同じ技術を持っているために主客の転倒が頻繁に起こるのだろう。このような場のルールが頻繁に変わるなかで技術的なトレンドを理解しつつプログラミングを学習していくのが重要なコツとなっている。非常にオープン指向でありながら一方で内輪の流行などにも左右されており、デザインする側とさ



れる側が入り乱れる Web 技術の学習環境は歪んだ情報生態系であるとも言えるだろう。

また、リアルでのゲークたちの活動を見ていく中で、テクノロジーを社会的な規律を維持するために使うような指向性は少ないと言えるのではないだろうか。テクノロジーを生かして自由に奔放な活動ができるようにその質を高めるという実践が数多く行われているのである。さらに、海外の一部のゲークは監視技術に対して技術的な反抗を試みるという活動も生まれつつある。

今回の一連のフィールドワークによって、ゲークのコミュニティが技術的に極端に先進的なユーザーと接触できる機会として貴重な場であることが明らかとなった。そのため今後の展望としては、こうしたエクストリーム・ユーザー = ゲークたちのコミュニティが新たなテクノロジー発掘したり、既にあるテクノロジーを再デザインする上でのアイデアを得ることができる場として捉え、より深くエスノグラフィーを展開していく。そうした深いエスノグラフィーを可能とする過程として、私自身が先述した API やマッシュアップ事例のように、ゲークコミュニティを再編・再デザインできるようなサービスを開発することも必要となってくると思われる。

## 参考文献

- [ 1 ] 上野直樹, 「仕事の中での学習 - 状況論的アプローチ」, 東京大学出版会(1999/11)
- [ 2 ] 上野直樹, 土橋臣悟, 「科学技術実践のフィールドワーク - ハイブリッドのデザイン」, せりか書房(2006/12)
- [ 3 ] 上野直樹, ソーヤーりえこ, 「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(2007/03)
- [ 4 ] 梅田望夫, 「ウェブ進化論? 本当の大変化はこれから始まる」, 筑摩書房(2006/02)
- [ 5 ] 宮川達彦, 「Bulkfeeds とゲークコミュニティ」, [http://japan.cnet.com/blog/umeda/2004/11/24/entry\\_bulkfeeds/](http://japan.cnet.com/blog/umeda/2004/11/24/entry_bulkfeeds/)
- [ 6 ] 宮川達彦, 「アルファゲークと Hack」, [http://japan.cnet.com/blog/umeda/2004/11/25/entry\\_hack/](http://japan.cnet.com/blog/umeda/2004/11/25/entry_hack/)
- [ 7 ] Brad Fitzpatrick 著, kentaro 訳, 「ソーシャルグラフについて」  
<http://d.hatena.ne.jp/antipop/20070819/1187527599>
- [ 8 ] Social Graph API - Google Code, <http://code.google.com/apis/socialgraph/>
- [ 9 ] Greasespot, <http://www.greasespot.net/>
- [ 10 ] Userscripts.org, <http://userscripts.org/>
- [ 11 ] Coderepos, <http://coderepos.org/share/>
- [ 12 ] Favotter, <http://favotter.matope.com/>