

ノウアスフィアの開墾

(*Homesteading the Noosphere*)

Eric S. Raymond *

山形浩生† 訳

1998年7月、1998年7月訳、2000年5月1日更新

概要

オープンソースのライセンスで定義された「公式」イデオロギーとハッカーたちの実際の行動には矛盾が観察される。これをふまえて、ぼくたちはオープンソースソフトの所有権とコントロールをめぐる実際の慣習を検討する。そこで明らかになったのは、そうした慣習の根底にあるのが、ロックの土地保有に関する理論と類似した、所有権の理論であるということだ。これと関連づけるかたちで、ハッカー文化を「贈与文化」として分析する。つまりその参加者たちは時間とエネルギーと創造性をあげてしまうことで、名声を競うわけだ。さらにこの分析が、ハッカー文化における紛争解決にとってどのような意味を持つかを検討し、いくつかの処方箋的な示唆を得るものとする。¹

目次

1	そもそもの矛盾	2
2	ハッカーイデオロギーのさまざま	3
3	放縦な理論と純潔な実践	5
4	所有権とオープンソース	6
5	ロックと土地所有権	9
6	贈与経済としてのハッカー文化	11
7	ハッキングのよろこび	12
8	評判のさまざまな相貌	13
9	所有権と評判によるインセンティブ	14

*<http://www.tuxedo.org/esr/>

†<http://www.post1.com/hiyori13/>

¹©2000 Yamagata Hiroo. この著作権表示を残す限りにおいてこの翻訳は商業利用を含む複製、再配布が自由に認められる。プロジェクト杉田玄白 (<http://www.genpaku.org/>) 正式参加作品。
最新版は <http://cruel.org/freeware/noosphere.html> を参照のこと。

10 エゴの問題	16
11 謙虚さの美德	17
12 評判ゲームモデルが持つグローバルな意義	19
13 どれほどすばらしい贈り物？	21
14 ノウアスフィア的所有権となわばりの動物行動学	23
15 紛争の原因	24
16 プロジェクト構造と所有権	25
17 紛争とその解決	26
18 文化への順応過程とアカデミズムとの関連	27
19 結論：慣習から慣習法へ	30
20 これからの研究の課題	31
21 書誌、注	31
22 謝辞	36
23 改訂履歴	36

1 そもそも矛盾

インターネットのオープンソースソフトの、活発でものすごく生産的な世界をしばらくでもながめていれば、だれでもオープンソースハッカーが信念として語る内容と、かれらの実際の行動との間におもしろい矛盾があることに気がついてしまうだろう。つまり、オープンソース文化の公式イデオロギーと、その実際の活動は一致していないわけだ。

文化は適応力のある機械だ。オープンソース文化は、はっきりした動きや圧力への反応として生まれてきている。いつもながら、ある文化の状況への対応方法は、意識的なイデオロギーとしての部分と、無意識的またはあまりはっきり意識されない暗黙の知識としての部分との両方の形であらわれてくる。そしてよくあることだが、無意識的な適応は、意識されたイデオロギーと部分的に矛盾するところがある。

この論文では、この矛盾の根っこのまわりを掘り下げてみて、それをもとにこの文化を動かす動きや圧力を明らかにする。そしてハッカー文化とその習慣について、おもしろい事実を導き出す。

そして結論として、この文化が内に持った知識をもっといいほうに活用するための方法をいくつか示唆してみる。

カッコに入ったラベルは、参考文献と巻末注をさしている。

2 ハッカーイデオロギーのさまざま

インターネットのオープンソース文化のイデオロギー（ハッカーが信念として口にするもの）は、それ自体がなかなかややこしい代物ではある。メンバーはみんな、オープンソース（つまり自由に再配布ができて、ニーズの変化に対応して自由に発展・変更可能なソフト）がいいものであって、大がかりに力をあわせる価値のあるものだということには合意している。この合意がこの文化への参加を実質的に定義つけるものだ。でも、この信念の理由として各個人やさまざまな下位の小文化群が挙げるものは、かなりちがっている。

大きな差がみられるのは熱狂性だ。オープンソースがある目的（よいツールやおもしろいおもちゃや楽しいゲーム）を実現するための、ただの便利な手法としてとらえられているか、それともそれ自体が目的として考えられているか。

すごく熱狂的な人間なら、こんなことをいうだろう。「フリーソフト命！ ぼくが存在しているのは、便利で美しいソフトや情報リソースをつかってそれをあげてしまうためなんだ！」 そこそこの熱狂性をもった人間なら「オープンソースはいいことだし、だからその実現のために相当量の時間を喜んで割きますよ」というだろう。そんなに熱狂的でない人間なら、こういうかもしれない。「うん、オープンソースはいいときもあるわよね。たまにいじってみたりもするし、それをつくった人たちには敬意を払ってるわ」

ほかに差があるところは、商用ソフトへの敵意と、商用ソフト市場を支配しているとみられる企業への敵意だ。

すごく反商業主義的な人間はこういうだろう。「商用ソフトは窃盗行為で秘匿行為だ。ぼくがフリーソフトを書くのはこの悪行を止めさせるためだ」。そこそこの反商業主義的な人間はこういう。「商業ソフトそのものはおおむね OK よ。だってプログラマには対価が支払われるべきだもん。でも、いい加減な製品にあぐらをかいて、数をたのみにごりおししてまわる企業はろくでもないわね」そして反商業主義でない人間ならこういう。「商用ソフトだってよいではないの。ぼくがオープンソースソフトを書く / 使うのは、そっちが好きだからってだけよ」

以上のカテゴリーをかけあわせて出てくる 9 つの態度すべてがオープンソース文化の中には登場している。こういうちがいを指摘しておくことが大事なのは、それが異なった目的を意味するからで、したがって適応行動や協調行動もちがってくるからだ。

歴史的にいうと、もっとも目に見えて、いちばんよく組織化されていたハッカー文化の部分は、きわめて熱狂的で、非常に反商業的だった。リチャード・M・ストールマン (Richard M. Stallman, RMS) の創設したフリーソフトウェア財団 (Free Software Foundation, FSF) は 1980 年代初期以来、大量のオープンソース開発を支援してきた。そのなかには、Emacs や GCC のような、イン

ターネット・オープンソース界にとって、いまでも、そしてこの先当分の間も、基本となるツールであり続けるようなものも含まれている。

何年ものあいだ、FSF はオープンソース・ハッキングに唯一最大の焦点だったし、いまだにこの文化にとってきわめて重要なツールを生み出し続けてきている。FSF はまた、ハッカー文化を外から見ている者にとっては、ながいことオープンソースを支援している組織的な身分をもった唯一のスポンサーだった。かれらこそ実質的に「フリーソフト」という用語を定義し、このことばに意図的に挑戦的な含みを持たせるようになった。(新しいラベルの「オープンソース (<http://www.opensource.org/>)」は同じくらい意図的にそういう含みをさけるようにしている。)

だからハッカー文化に対する認識は、内からも外からも、FSF の熱狂的な態度と反商業主義的な目標と同一視されがちだった (RMS 自身は、自分が反商業主義であることを否定している。でもかれの活動は、そのもっとも活発な支持者たちを含め、ほとんどの人からそのように受け取られている)。FSF の「ソフトウェアの隠匿をつぐせ！」という精力的ではっきりした運動は、ハッカー・イデオロギーにいちばん近いものとなったし、RMS はハッカー文化のリーダーにいちばん近い存在となった。

FSF のライセンス条項「General Public License (GPL)」は FSF の熱狂的で反商業主義的な態度を表現している。これはオープンソース界では広く使われているライセンス方式だ。ノースカロライナ大の Sunsite は、Linux 界で最大かついちばん人気のあるソフトウェアアーカイブだけれど、1997 年 7 月には Sunsite のソフトパッケージではっきりしたライセンス条項を持つもののうち、半数が GPL を使っている。

でも、FSF はいつの時点でも決してこの世界唯一の団体なんかじゃなかった。ハッカー文化にはつねに、もっと静かで挑戦性の低い、もっと市場を敵視しない流れがあった。このプラグマティストたちは、イデオロギーよりはむしろ、FSF に先立つ初期のオープンソース的な動きに基づいた工学的な伝統に対して忠誠を感じていた。この伝統でいちばん重要なものとしては、Unix と商業主義以前のインターネットの、相互にからみあった技術文化が含まれる。

典型的なプラグマティストの態度は、ごく穏健な反商業主義で、それが企業世界に対して持っている不満の大部分は「秘匿」そのものじゃあない。むしろそれは、企業世界が Unix やオープン規格やオープンソースなんかを採用している優れたアプローチをなぜか拒むという点だ。プラグマティストたちが嫌うのは何かといえば、それは「隠匿者たち」一般よりはむしろ、ソフトウェア主流派の目下の帝王だろう。これはかつては IBM だったし、いまはマイクロソフトだ。

プラグマティストたちにとって、GPL はそれ自体が目的なんじゃなくて、ツールとして重要だ。その主な価値は「秘匿」に対する武器としてのものじゃない。むしろソフトの共有とバザール様式 (<http://cruel.org/freeware/cathedral.html>) の開発コミュニティ成長を奨励するためのツールとして大事になる。プラグマティストは、商業主義を嫌うよりはいいツールやおもちゃを手に入れるほうを重視するし、商業製品を使う場合でも、それが高品質なら別にイデオロギー上の不快感は感じない。同時に、オープンソース界での経験でおぼえた技術的な品質は、閉鎖的なソフトではほとんど実現困難なほど高い。

何年にもわたって、ハッカー文化の中のプラグマティスト的な視点は、特に GPL を完全に受け入れたり、あるいは FSF の目標一般を受け入れることを頑固に拒否する流れとして己を主張しつつづけていた。1980 年代と 1990 年代初期には、この態度はパークレー Unix のファンや BSD ライセンスの使用者、そして BSD ソースをもとにオープンソースの各種 Unix をつくろうという初期の努力と結びつけられていた。でもこうした試みは、しかるべき規模のバザール・コミュニティ形成に失敗し、きわめて断片化されて力の弱いものとなってしまった。

プラグマティストたちが本物の勢力基盤を見つけるには、1993 年初期から 1994 年にかけての Linux の爆発を待たなくてはならなかった。Linus Torvalds は決してあえて RMS に反対したりはしなかったけれど、でも商業 Linux 産業が成長するのをだまってみ過ごし、特定の仕事には高品質な商業ソフトの利用を推奨し、そしてハッカー文化のもっと純粹主義で狂信的な部分を軽く嘲笑することで、先例を確立したわけだ。

Linux 急成長の副作用として、新しいハッカーたちが多数登場したことが挙げられる。かれらにとって、主な忠誠は Linux に対してのものであって、FSF の目標はもっぱら歴史的興味でしかなかった。Linux ハッカーのもっと新しい波は、Linux システムを「GNU 世代の選択」と表現はするけれど、でもその多くはストールマンよりはトーヴァルズをまねがちだ。

しだいに、むしろ反商業的な純潔主義者たちのほうが少数派においやられていった。事態がどれだけ変わったかがはっきりしたのは、1998 年 2 月に Netscape 社が Navigator 5.0 をソースコードで配布すると発表したときだった。これは企業世界での「フリーソフト」に対する興味をかきたてることになった。これに続いてハッカー文化に対し、このかつてない機会を利用しつつともに、その成果を「フリーソフト」から「オープンソース」と命名しなおそうという呼びかけが行われたけれど、これに対しては即座にもすごい支持が得られたので、関係者はだれしもびっくりしたほどだ。

これをさらに補強する展開として、ハッカー文化のプラグマティスト部分そのものも、1990 年代半ばにはだんだん多心型になっていった。Unix/インターネットの根っこの株からは、ほかにも半独立のコミュニティが生まれるようになり、それぞれが独自の自意識とカリスマ的なリーダーを持っている。なかでも Linux 以降でいちばん必要なのは、Larry Wall 率いる Perl 文化だ。もっと小さいながらも重要なものとしては、John Osterhout の Tcl や Guido Van Rossum の Python 言語を取り巻く伝統が挙げられる。この三つとも独自の GPL でないライセンス方式を編み出すことで、イデオロギー的な独立性を主張している。

3 放縦な理論と純潔な実践

とはいえこうした変化すべてを通じて、「フリーソフト」や「オープンソース」がなんであるかについては、おおまかな合意を得た理論が維持され続けていた。この共通理論のいちばんはっきりした表現は、いろんなオープンソース・ライセンスにみることができる。そのすべてにはきわめて重要な共通要素があるんだ。

1997年に、こういう共通要素が蒸留されて Debian フリーソフトウェア・ガイドラインとなり、これが Open Source Definition (<http://www.opensource.org/>) になった。OSD の定義したガイドラインのもとでは、オープンソースのライセンスはオープンソース・ソフトを変更（そしてその変更版を再配布）する権利をだれにでも無条件に認めなくてはならない。

だから OSD（そして GPL、BSD ライセンス、Perl の Artistic License など OSD 対応のライセンス群）の暗黙の理論というのは、だれもがなんでもハックできるというものだ。どんなオープンソース製品（たとえば FSF の gcc C コンパイラとか）でも、6 人くらいがソースを複製してそれを勝手にちがった発展方向に向けて変えてしまい、しかもみんなが「これぞ本家本元」と主張する、といったことは十分可能だし、それを止めるものはなにもない。

でも実際には、こんな「分裂」はほとんど絶対に起きないといっていい。大プロジェクトの分裂はすごくまれだし、いつも名前が変わるうえ、世間的にもなぜそれが正当かについて、大量の説明が行われる。GNU Emacs/XEmacs の分裂や、さまざまな BSD 系グループの分裂なんかでは、分離はたちは自分たちが非常に強力なコミュニティ規範に逆らって行動していると感じていたのは明らかだ。[SP]

実際問題として（そして「だれでもなんでもハックできる」という合意があるという理屈とは正反対に）、オープンソース文化は入念ながらほとんど認識されていない所有権の慣習を持っている。この慣習によって、だれがソフトを変更できるか、どういう状況でそれが変更できるか、そして（特に）だれが変更バージョンを再配布してコミュニティに戻せるか、が規定されているんだ。

ある文化の禁忌（タブー）は、そこでの規範をシャープに浮かび上がらせる。だから、だいたいなタブーをここでまとめておくと、あとあと役にたつだろう。

- プロジェクトの分岐に対してはすごく強い社会的な圧力がある。どうしてもこれが必要なんだという請願のもとで、世間に対してもその行為を正当化する訴えがたくさん行われ、そしてプロジェクトの名前も変えない限り、それは起こらない。
- プロジェクトへの変更を、モデレータたちの協力なしに行うといい顔をされない。ただし、基本的に些末な移植上のフィックスなどはのぞく。
- ある人の名前をプロジェクトの歴史やクレジットや管理者リストからのぞくのは、当人のはっきりした合意なしには絶対に行われぬ。

この論文ではこれから、こうしたタブーと所有についての慣習を検討する。それらがどう機能するかというだけでなく、それが明らかにしているオープンソース・コミュニティの根底にある社会的な力学やインセンティブ構造についても見ていこう。

4 所有権とオープンソース

ある物件が無限に複製可能で、いくらでも変えられて、そしてそれをとりまく文化が脅せるような権力関係や財の希少性に基づく経済に基づいていないとき、なにかの「所有権」を持つていう

のはどうということだろうか。

実はオープンソース文化に関するかぎり、この質問には簡単に答えられる。ソフトウェア・プロジェクトの所有者というのは、変更したバージョンを公式に再配布する独占的な権利をコミュニティ全般から認められている人物である。

(この章で「所有権」を論じるときには、その所有者については単数形を使う。これだとまるですべてのプロジェクトは一人の人物が所有しているかのようだけれど、プロジェクトをグループで所有している場合もあるということは承知してほしい。こういうグループ内の力学についても、この論文で見えていくことにする。)

標準的なオープンソースのライセンスのもとなら、ソフト進化のゲームではみんなが平等なはずだ。でも現実には、公認の管理者が承認して進化するソフトに統合した「公式」パッチと、第三者による「非公式」パッチとははっきり区別して認識されている。非公式パッチはあまりないし、一般には信用のおけないものとされる。[RP]

公式な再配布こそが根本的な問題なんだというのは、簡単に示せる。ハッカー文化の慣習は、みんなが自分だけのためにソフトをパッチするのは推奨している。この慣習は、閉じたユーザグループや開発グループ内で変更バージョンを再配布する人々も問題にはしない。そういう変更がオープンソース・コミュニティ一般に対してポストされ、オリジナル版と競合するようになったときにだけ、所有権が問題になってくる。

一般的にいて、オープンソース・プロジェクトの所有権を獲得するには3つの方法がある。第一の方法はいちばん自明だろうけれど、プロジェクトを創始することだ。プロジェクトの開始以来、管理者が一人しかいなくて、その管理者が活動を続けているなら、ハッカー慣習はそのプロジェクトをだれが所有しているのかについて疑問視することすら許さない。

プロジェクトの所有権を獲得する第二の方法は、前の所有者からそれを引き継ぐことだ(いわゆる「バトンタッチ」)。プロジェクトの所有者は開発や保守作業に必要な時間を割けなかったり割く気がなかったりするときには、有能な後継者にプロジェクトを引き継ぐ義務がある。これはこのコミュニティではよく理解されている。

大きなプロジェクトの場合、こういうコントロールの引き継ぎは一般に派手なファンファーレつきでアナウンスされていることはだいじだ。オープンソースコミュニティ全体として、所有者の後継者選びに実際に文句がついた例はまったくないけれど、慣習的な行為は明らかに、世間的に見た正当性がだいじなんだという認識を含んでいる。

小規模なプロジェクトでは、プロジェクトの配布パッケージの中に変更履歴を含めておいて、そこで所有者の変化を明記しておけばふつうは十分だ。ここでの明らかな過程は、もし前の所有者が自発的にコントロールをゆずったのでなければ、その人はそれなりの期間内に公開の場で抗議することで、コミュニティの支援のもとでコントロールを奪回できる、というものだ。

所有権を獲得する第三の方法は、そのソフトに作業が必要だとみてとって、しかももとの所有者が消えたか興味を失ったかしたときだ。もしこれをやるなら、所有者を見つけようとするのがきみの責任だ。それがうまくいかなければ、しかるべき関係した場所(たとえばそのアプリケーション

分野専門の Usenet ニュースグループなんか)でそのプロジェクトがどうも放棄されたらしくて、だから引き継ごうと思うんだけど、と宣言することになる。

慣習は、自分が新しい所有者だと宣言するまえに、しばらく様子を見ると要求している。このあいだに、だれかが実は自分はそのプロジェクトで作業を続けるんだけど、と申し出れば、その人たちの申し立てが優先されることになる。プロジェクトを引き継ぎたいという意志発表は、何回かやるのがいいとされる。それも、複数の場でやったほうがポイントは高い(関連ニュースグループ、メーリングリストなど)。そして返事を待つのに辛抱強ければもっといい。一般に、前の所有者やほかの権利保持者が反応できるようになるべく努力をしたほうが、反応がないときの自分の権利主張が通りやすい。

このプロセスを、プロジェクトのユーザコミュニティにも見えるところでやって、反対がなければ、その捨て子プロジェクトの所有権を主張して、その旨を履歴ファイルに記入できる。でもこの方法は、ボタンタッチされるよりも危ういもので、ユーザコミュニティから見て相当な改良を加えない限り、正式に認められたと思っちゃいけない。

ぼくはこういう慣習が実際に働くのを、FSF 以前のオープンソースの古代にさかのぼる 20 年前から見てきている。これにはいくつかともおもしろい性質がある。なによりもおもしろいことの一つは、ほとんどのハッカーたちは、実際に自分がそうしていると完全に意識していないのに、これにしたがってきたということだ。この文章は、実際に文になった意識的でそれなりに網羅的なまとめとしては、史上初のものかもしれない。

もう一つ挙げられるのは、無意識の慣習だというのに、それはきわめて(それどころか驚異的なほど)一貫して守られてきている、ということだ。ぼくは文字通り何百ものオープンソースプロジェクトの進化を見てきた。それなのに、見聞きした大きな逸脱行為はまだ両手の指で数えられるほどしかない。

でも、第三のおもしろい性質というのは、こういう慣習が時を追って発展してくるとき、それは一貫した方向性を持っていたということだ。この方向性というのは、もっと公共的なアカウントビリティや通達を奨励し、プロジェクトのクレジットや変更履歴の保存についてもっと配慮を増やして(なににもまして)いまの所有者の正当性を確立しようというものだった。

こういう性質が示唆しているのは、こうした慣習が群善の産物ではなくて、オープンソース文化でのなにか暗黙の目的や生成パターンの産物で、その機能方法にとってきわめて根本的なものなんだということだ。

この論文に早い時期にコメントをよせたある人は、インターネット・ハッカー文化とクラッカー・海賊文化(ゲームのクラッキングと海賊ソフト BBS を中心に活動する、「Warez D00dz」)を対比させると双方の生成パターンがなかなかはっきりすることを指摘してくれた。d00dz との対比は後出²。

²訳注: warez d00dz 「ウェアズ・デューズ」と読む。Warez は wares であり、海賊コピーされたソフト群など、この種の BBS 上でやりとりされるソフト一般をさす。D00dz は doodz であり、これは dude の複数形を発音に忠実に表記したもの。敢えて訳すと「(ソフト)ウェア野郎ども」かな。濁音になる S をすべて Z で置き換え、さらに o(オー)の字を数字のゼロで置き換える独特の表記に注目。かれらの BBS では多くの人がこのようなハンドルを使用する。A と似ているので 4 を使ったり、for のかわりに 4 を使ったり、大文字小文字を逆転させることも多い。ありとあらゆるソフ

5 ロックと土地所有権

この生成パターンを理解するには、ハッカーの通常の関心領域をはるかに離れて、こうした慣習と歴史的に類似した例を見るのが役に立つ。法制度史や政治哲学を学んだ人ならお気づきのように、ハッカーたちが暗に主張している所有理論は、英米慣習法における土地所有権の理論とほとんどまったく同じなんだ！

この理論では、土地の所有権を獲得する方法は3つある。

未開の地（フロンティア）には、これまで所有者のいなかった土地がある。そこでは人は、開墾（homesteading）することで所有権を獲得できる。つまり、自分の労働を所有されていない土地に混ぜ込み、囲いをつけて自分の地権を守ることによって。

入植済みの地域での土地移譲は、ふつうは土地所有権の移転によって行われる。これはつまり、前の所有者から証書を受け取ることだ。この理論に基づけば、所有権の連鎖という概念が重要になる。所有権の証明として理想的なのは、証書とその移転の連鎖が、その土地のそもそもの開墾時点にまでさかのぼれることだ。

最後に、慣習法理論は土地の所有権が失われたり放棄されたりすることがあるのを認識している（たとえば所有者が相続人なしに死んだり、空き地に対する所有権の連鎖を確立するための記録が失われていたりする場合）。このようにして遺棄された土地は、占拠によって所有権の主張を行える。だれかがそこに入居して、整備を行い、そして開墾時と同じようにその所有権を防衛するわけだ。

この理論はハッカーの慣習と同じように、中央権力が弱いか存在しないような場面で有機的に発達してきた。それは千年以上もかけて、ノルウェイやドイツの部族法から発展してきたものだ。それを近代の初期に体系化して合理化したのがイギリスの政治哲学者ジョン・ロックだったので、これはよく所有物の「ロック」理論と呼ばれる。

論理的によく似た理論は、ある物件が経済的または生存上で高い価値を持っていて、しかも稀少財の配分について、中央集権的に強制するだけの力を持った単一の権威が存在しないところでは必ず生じるようだ。これは、ときどきロマンチックに「所有」の概念がないと考えられたりもする狩猟採集文化でも成り立つ。たとえばKgalagadi（カラハリ）砂漠のクンサン・ブッシュマンの伝統では、狩猟場には所有権は存在しない。でも、井戸や泉については所有権が存在して、その理論は明らかにロックの理論と似ている。

クンサン・ブッシュマンの例は示唆的だろう。というのも、これはロック式の所有権慣習が、その資源からの期待リターンがそれを独占して守る期待コストよりも大きい場合にのみ生じることを示してくれるからだ。狩猟場が所有権の対象にならないのは、狩猟のリターンがすぐ予想しにくくて変動しやすく、さらに（すごく高く評価はされるけれど）日々の生存にとって絶対的に必要なものではないからだ。一方の井戸は、生存にとって不可欠だし、守りきれぬくらい小さい。

この論文の題名に出てくる「ノウアスフィア」というのはアイデア（観念）の領域であり、あ

トが海賊版で流通しているが、d00dzの多くは別にそれを使うわけではなく、単に保持タイトルを自慢しあっているケースが多い。

らゆる可能な思考の空間だ。[N] ハッカーの所有権慣習に暗黙に含まれているのは、ノウアスフィアの部分集合の一つであるすべてのプログラムを包含する空間での、所有権に関するロック理論なんだ。だからこの論文は「ノウアスフィアの開墾」と名付けた。新しいオープンソース・プロジェクトの創始者がみんなやっているのがそれだからだ。

Fare Rideau (rideau@ens.fr) は、ハッカーは純粹觀念の領域だけで活動しているとはいえないと指摘している。これは正しい。かれの主張だと、ハッカーが所有しているのはプログラミング・プロジェクトなんだ。これはつまり、物質的な労働（開発、整備など）の意図的な焦点で、ここに評判、信用性なんか結びついている。ここからかれは、ハッカーのプロジェクトが覆っている空間はノウアスフィアではなく、むしろその分身のような、ノウアスフィアを探索するプログラム・プロジェクトだと主張する（読者のなかの天文物理学者たちに敬意を表しつつも、語源的に言えばこの分身空間を「エルゴスフィア」または「作業のスフィア」と呼んでまちがいない）。

現実問題としては、ノウアスフィアとエルゴスフィアとの区別はこの論文の目的からいえばだいじじゃない。Fare が主張するような純粹な意味での「ノウアスフィア」が、なんら意味あるかたちで存在すると言っていいのかは、実はよくわからない。そんなものを信じるには、プラトン主義哲学者にでもなるしかないだろう。そしてノウアスフィアとエルゴスフィアとの区別が実際の重要性を持つのは、アイデア・觀念（ノウアスフィアの要素）は所有できずそれをプロジェクトとして実体化したものだけが所有できると考えたい場合だけだ。この問題から出てくる課題は、知的所有権の理論的な問題につながるもので、この論文の範囲を超える。[DF]

でも混乱を避けるために言っておくと、ノウアスフィアもエルゴスフィアも、時に（多くのハッカーたちは顔をしかめるけれど）「サイバースペース」と呼ばれる電子メディア内での仮想位置全体とはちがう。サイバースペースでの所有権はまったくちがった規則にしたがって統制されている。その規則は物質的な世界での規則のほうに、むしろ似ている。基本的には、その「サイバースペース」の一部を宿しているメディアやマシンを所有している人物が、そのサイバースペースを結果的に所有していることになる。

ロック的な構造は、オープンソースのハッカーたちがその慣習をまもるのは、なにか自分たちの努力からの、一種の期待収益を守ろうとしているからだろうと強く示唆している。この収益は、プロジェクト開墾の努力や「所有権の連鎖」を記録したバージョン履歴を維持するコスト、そして捨て子になったプロジェクトを占拠してするまでに公的な通達を出してしばらく待つという時間コストよりもずっと大きなものでなくてはならない。

さらに、オープンソースからの「収益」は単なるソフトの利用価値以上のものだけでなきゃならない。プロジェクトの分裂によってダメになったり減ったりしてしまうようなものでなければ。もし利用価値だけが問題なら、分裂に対するタブーはないし、オープンソースの所有権も土地所有権とは似てもつかないはずだ。事実、既存のオープンソース・ライセンスがから導かれるのは、こういう（利用価値だけが唯一の収益となる）別世界だ。いくつか候補としてあがる収益はすぐに却下できる。ネット接続上では有効な脅しがかげられないので、権力追求はまったく成り立たない。同じく、オープンソース文化はお金や内的な稀少性経済に類するものは持っていないから、

ハッカーたちも物質的な富とあまり似たものを追求していることはあり得ない。

ただし、オープンソース活動で人々がもっと金持ちになれる方法が一つなくはない。そしてそれは、その実際に動機に貴重なヒントを与えてくれるものではある。しばしば、ハッカー文化で人が獲得した評判は実世界でも反映されて、それが経済的に意味をもってくることがある。もっといい仕事が得られるとか、コンサルタント契約が手に入るとか、あるいは本の執筆依頼がくるとか。

でもこの種の副作用は、よくいってもまれだし、ほとんどのハッカーにとっては副次的なものでしかない。唯一の説明として説得力を持つには、多くのハッカーにはあまりに縁遠いものだ。それにハッカーたちはなんども、自分たちは金のためにやってるんじゃない、理想と愛のためにやってるんだ、と主張している。まあこれは割り引いてきくにしても。

でも、こういう経済的な副作用が処理されるやり方は検討する価値がある。これから、オープンソース文化そのもののなかでの評判の力学だけでも、かなりの説明力があることを見てみよう。

6 贈与経済としてのハッカー文化

オープンソース文化での評判の役割を理解するには、歴史から離れて文化人類学と経済学に深入りし、交換の文化と贈与の文化のちがいを検討してみると役に立つ。

人間は、社会的地位のために競争しようという生まれながらの衝動がある。それはヒトの進化の歴史のなかでからだに刻み込まれているんだ。その歴史のうち、農業の発明に先立つ90年くらい暮らしていた。地位の高い個人は、一番健康な伴侶と最高の食料へのアクセスを手に入れた。この地位への衝動は、おもに生存に必要な財の稀少性の度合いに応じて、いろいろな形であらわれてくる。

人間が持つ組織化のほとんどの方法は、希少性と欲求に対する適応行動だ。それぞれの方法は、社会的地位を獲得する別々の手段を持っている。

一番簡単な方法は上意下達方式 (command hierarchy) だ。上意下達方式では、希少な財の配分は一つの中央権力が行って、それが軍事力でバックアップされる。上意下達方式は、規模の変化への適応力 (スケラビリティ) がものすごくとぼしい [Mal]。大きくなるにつれて、ますます横暴で非効率になってゆく。このため、大家族以上の上意下達方式はほぼかならずといっていいほど、別のかたちのもっと大きな経済に寄生する存在でしかない。上意下達方式では、社会的地位はおもに恐喝力へのアクセス能力によって決まってくる。

ぼくたちの社会はもっぱら交換経済だ。これは財の希少性に対する洗練された適応方式で、規模の変化にもよく適応する。希少な財の配分は、交換と自発的な協力によって非中心的に行われる (そして実は、競争の欲望がもたらす最大の効果は協力的行動を生み出すことだ)。交換経済では、社会的地位はおもにもの (必ずしも物質的なものとは限らない) のコントロールの大小で決まる。

ほとんどの人は、この二つについては説明されるまでもなく精神的なモデルを持っているし、それらがどう相互に機能するかもわかっている。政府や軍、ギャング集団などは、ぼくたちが「自由市場」とよぶもっと大きな交換経済に寄生している上意下達システムだ。しかしながら、このどち

らともまったくちがっていて、人類学者たち以外はあまり認知されていない第三のモデルがあるんだ。これが贈与の文化だ。

贈与文化は、希少性ではなく過剰への適応だ。それは生存に不可欠な財について、物質的な欠乏があまり起きない社会で生じる。穏和な機構と豊富な食料を持った経済圏の原住民の間には、贈与と経済が見られる。ぼくたち自身の社会でも、一部の層では観察される。たとえばショービジネスや大金持ちの間でだ。

過剰は上意下達関係を維持困難にして、交換による関係をほとんど無意味なゲームにしてしまう。贈与の文化では、社会的なステータスはその人がなにをコントロールしているかではなく、その人がなにをあげてしまうかで決まる。

だからクワキトルの酋長はポトラッチ・パーティーを開く。億万長者は派手にフィランソロフィー活動をして、しかもそれをひけらかすのが通例だ。そしてハッカーたちは、長時間の労力をそそいで、高品質のオープンソース・ソフトをつくる。

というのも、こうして検討すると、オープンソース・ハッカーたちの社会がまさに贈与文化であるのは明らかだからだ。そのなかでは「生存に関わる必需品」つまりディスク領域、ネットワーク帯域、計算能力などが深刻に不足するようなことはない。ソフトは自由に共有される。この豊富さが産み出すのは、競争的な成功の尺度として唯一ありえるのが仲間内の評判だという状況だ。

でもこの観察は、それだけではハッカー文化に見られる特徴を説明するのに十分とはいえない。クラッカー d00dz は、ハッカーと同じ（電子）メディア上に息づく贈与文化を持っているけれど、その行動は大きくちがっている。このグループの性行は、ハッカーよりもずっと強く排外的だ。共有するよりは秘密を隠匿したがる。クラッカーグループでは、ソフトクラック用のソースのない実行ファイルが出回ることのほうが、そのやりかたを教えるヒントが出回ることより多い。（この行動についての、d00dz 集団内部からの観点については [LW] を参照。）

これが示しているのは、もちろん言うまでもなく自明だろうけれど、贈与文化の運用方法は一つじゃないってことだ。ぼくはハッカー文化の歴史について、別のところでまとめたことがある。[HH] それがいまの行動を形成していった方法は、謎なんかじゃない。ハッカーたちは、自分たちの文化を規定するにあたり、自分たちの競合が行われる形式の集合を利用したわけだ。この論文ではこの先、その形式を検討することにしよう。

7 ハッキングのよろこび

ついでに言うておくと、この「評判ゲーム」の分析をするからといって、別に美しいソフトを設計してそれをうまく動かすことの芸術的な満足を軽んじたり、無視したりするつもりはない。ぼくたちみんな、こういう満足は経験しているし、そのために生きているようなところさえある。そういうのがだいじな動機になっていないような人間は、そもそもハッカーになつたりはしない。ちょうど、音楽を好きでない人間が作曲家になつたりしないのと同じことだ。

だったら、職人氣質の純粋なよこびが主な動機となっているハッカー行動のモデルを考えたほうがいいのかな？ この「職人氣質」モデルは、職人氣質発揮機会最大化とその結果の最大化を達成する手段としてハッカー慣習を説明できなくてはならない。これは、「評判ゲーム」モデルと対立したり、別の結果を示唆したりしているだろうか？

そうでもない。「職人氣質」モデルを検討すると、結局はハッカー社会を贈与文化のように動かしている制約条件と同じ問題に戻ってくることになる。品質をはかる尺度がなければ、品質を最大化なんかできないだろう。希少性の経済が働かないなら、仲間内の評価以外の尺度なんかありえないだろう。どうも、職人氣質文化はけっきょくのところ、評判ゲームを通じて組織されるしかないんじゃないか。そして中世ギルド以来の多くの職人文化では、まさにこの力学が機能しているのがうかがえるんだ。

ある重要な一点で、「職人氣質」モデルは「贈与文化」モデルより弱い。それだけでは、この論文のきっかけとなったそもそもの矛盾は説明できない。

最後に、「職人氣質」的な動機はそれ自体として、ぼくたちが考えたがるほどは評判ゲームからかけ離れたものじゃないかもしれない。自分の書いたみごとなプログラムが引き出しに鍵をかけてしまいこまれていたらどうだろう。一方、それがうまく使われてみんなに喜ばれていたらどうだろう。どちらのほうが好きみとしては満足だろうか？

とはいっても、職人氣質モデルは捨てないでおこう。それは多くのハッカーにとって直感的にわかりやすいし、個人行動のある面はすごくうまく説明してくれるから。[HT]

この論文の初版を公開してから、匿名の評者がこう述べた。「意図的に評判を得ようとして作業はしないかもしれないけれど、でも評判というのは仕事をちゃんとしていれば、本当に利益をもたらす真の見返りなんだよ」。これは細かいが重要なポイントだ。評判インセンティブは、職人がそれを認識しているかどうかによらず、機能し続ける。だから究極的には、ハッカーが自分の行動を評判ゲームの一部として理解しているかどうかによらず、かれの行動はそのゲームに左右されることになる。

ほかの反応としては、同業者の中での名誉という報酬とハッキングの喜びが、生活維持に必要な水準以上になっているということ、人間の動機づけに関するアイブラハム・マズローの有名な「価値のヒエラルキー」モデルと関連づけている [MH]。この観点からすると、ハッキングの喜びというのは自己実現ないし高次ニーズであって、低次のニーズ（たとえば肉体的な安定性や「帰属感」や同業者の中での名誉など）が最低限は満たされない限り、一貫性を持った形で表明されることはないのだ、ということになる。だから評判ゲームは、ハッキングの喜びがまさに個人の主要動機となれるような社会的文脈を提供するにあたって、とても重要なものとなるかもしれない。

8 評判のさまざまな相貌

なぜ仲間内の評判（名声）が勝ち取るに足るものなのかについては、あらゆる贈与文化に共通する理由がある。

まず第一にいちばん明らかな点だけれど、仲間内のよい評判はそれ自体が重要な報酬だからだ。ぼくたちは、まえにふれた進化上の理由から、そう感じるようにつくられている（多くの人々は、名声に対する欲求をいろいろ昇華させて、はっきりした仲間グループと結びつかない「名誉」「倫理的誠実さ」「慈悲」などにリダイレクトすることを学ぶ。でもだからといって、根っこのメカニズムは変わらない）。

第二に、名声は他人の注目を集めて協力を得るのにすごく有効な方法だ（そして純粋な贈与経済では、唯一無二の方法だ）。もしある人が、気前よく、知的で、公正で、指導力があるとかいったよい資質で有名だったら、その人と関わりを持つことでメリットがあると他人に説得するのはずっと簡単になる。

第三に、もし贈与経済が交換経済や上意下達方式と接触していたり混じり合ったりしていた場合にも、評判がそっちに持ち越されて、もっと高い地位を得る役にたつかもれない。

こうした一般的な理由に加えて、ハッカー文化ではその特殊な条件のおかげで「現実世界」の贈与文化にくらべて名声がもっと価値の高いものになっている。

「特殊な条件」の最大のものは、ある人があげる成果物（あるいは別の解釈をするなら、その人の贈る労力と時間の目に見えるし）がとても複雑であることだ。その価値は、物質的な贈り物や交換経済のお金とはくらべものにならないほど不明確だ。よい贈り物とろくでもない贈り物を客観的に区別するのはずっとむずかしい。したがって、何かを贈る人が地位を求めて成功するかどうかは、仲間うちの批評的な判断に繊細に関わってくる。

もう一つ特異なのは、オープンソース文化が比較的純粋であることだ。ほとんどの贈与文化はまざりものが入っている。奢侈品の交易などで交換経済が入り込んでいたり、家族や部族集団みたいな上意下達関係が入っていたりする。オープンソースはこれに類するものはほとんどないと言っている。だから、仲間内の評判以外に地位を獲得する方法は、まずない。

9 所有権と評判によるインセンティブ

さてここまできたら、これまでの分析をひとまとめにして、ハッカーの所有権慣習をもっと統一的にとらえてみよう。ノウアスフィアの開墾からの収益はわかった。それはハッカーの贈与文化における仲間内の評判だ。そしてそれにとまなう二次的なメリットや副作用もついてくる。

この理解にもとづいて、ぼくたちはハッカー社会のロック流の所有権慣習が評判上のインセンティブを最大化する手段なんだと分析できる。つまり、仲間内のクレジットがしかるべき人物に帰属して、それ以外のところには行かないようにするわけだ。

これまで検討した3つのタブーは、この分析に基づけば完全に筋が通ってくる。他人が自分の作業を不当に横取りしたり、変な加工をしたりすれば、自分の評判が不当に落ちることになる。こういうタブー（そして関連した慣習）は、そういう事態を避けようとするものだ。

- プロジェクトの分裂がよくないのは、分裂以前に貢献していた人たちが評価上のリスクにさらされることになるからだ。このリスクをなくすには、かれらは分裂後の両方のプロジェク

トで同時に活動しなくてはならない（これは一般にややこしすぎたり、実践するのがむずかしすぎたりする）。

- 非公式パッチを配布する（あるいはもっとひどい場合には非公式バイナリを配布する）と、所有者たちは不公正な評判上のリスクにさらされる。公式コードが完璧であっても、所有者たちはパッチのバグのおかげで不評をかうはめになりかねない。（とはいえ、[RP]を参照のこと）。
- こっそりだれかの名前をプロジェクトからはずすことは、文化的な文脈では究極の犯罪だ。それは犠牲者の贈り物を盗み取って、泥棒のものとして提示するということからだ。

もちろん、プロジェクトをフォークさせたり非公式パッチを配布させたりするのは、もとの開発者グループの評判を直接攻撃することにもなる。もしぼくがきみのプロジェクトをフォークさせたり非公式パッチを配布したりすると、ぼくはつまりこう言っていることになる：「きみはまちがった決定を下した。（プロジェクトを、ぼくが導いているような方向に導いていない）」そしてぼくのフォーク版を使う人間はみんな、ぼくのきみに対する挑戦を支持していることになる。でもこれ自体は、フェアな挑戦だ。極端ではあるにしても。これは同業者（ピア）レビューの究極の形だ。だからこれは、これだけではタブーの説明としては不十分だ。それを強化するものであるのは確かだけれど。

このタブーの3つとも、被害者個人のレベルで害をもたらすとともに、オープンソースコミュニティに全体として害を及ぼすものだ。これらは、各潜在貢献者たちが、自分の贈り物・生産的行動に報酬が与えられるという可能性の認識を減らすことで、暗黙のうちに全コミュニティに害をなすことになる。

この三つのタブーのうち2つに対しては、別の説明候補があることは理解する必要がある。

まず、ハッカーはプロジェクト分裂に対する反感について、そんなことになったら子プロジェクトは当分はある程度似たような道をたどるから、作業が二重化して無駄だろ、と説明する。また分裂は共同開発者コミュニティを分裂させることが多く、だからどちらの子プロジェクトも、親より脳味噌が少なくなる点を指摘する人もいるだろう。

ある人のコメントだけれど、分裂の結果の子プロジェクトが、長期的にそれなりの「市場シェア」をもって一つ以上生き延びることは滅多にない。これはあらゆる関係者が協力して分裂を避けるインセンティブを強化する。だれが負け組に入るかを知るのはむずかしいし、負け組では自分たちの仕事の大半があっさり消えたり、無名なままで放置されたりすることになるからだ。

非公式パッチへの批判は、それがバグ追跡をもつごく難しくすることや、自前のバグを処理するだけでも手いっぱいな管理者の作業を、さらに増やす点などを観察したうえで説明されることが多い。

こうした説明には、かなりの真実が含まれている。そしていずれも、ロック式所有権の論理を強化するためにそれなりの働きをしてくれる。でも知的にはおもしろいけれど、これはタブーが曲げられたり破られたりするきわめてまれな状況で、なぜこれほどの感情となわばり性が発揮されるの

かという説明にはならない。しかもこれは、被害を受けた人々だけでなく、傍観者や見物人にもあてはまり、こういう人たちもかなり厳しい反応を示す。作業の二重化に関する冷たい計算や、管理の面倒の話などでは、こういう観察される行動はどうしても説明できない。

そしてさらに、第三のタブーがある。これを説明するのに、評判ゲーム分析以外にはなにも想像がつかない。このタブーが「そんなのフェアじゃないじゃん」という以上にはほとんど分析されなれないということは、それ自体が示唆的だ。これについては次の章で見る。

10 エゴの問題

この論文の冒頭で、ある文化の適応結果としての知識は、その意識的なイデオロギーと相反することも多い、と書いた。この大きな実例として、ロック式所有権慣習が、標準ライセンスに記載された意図に違反しているにもかかわらず、広く遵守されていることを見てきた。

この現象のおもしろい例が、評判ゲーム分析をハッカーたちと議論しているときにも観察できた。多くのハッカーたちはこの分析に難色を示し、自分たちの行動が仲間内の評判への欲望、あるいはその頃ぼくが浅い考えで使っていた名称では「エゴの満足」に動機づけられていると認めたらなかった。

これは、ハッカー文化についておもしろい点を示している。ハッカー文化は意識的にエゴ中心主義と、エゴに基づく動機づけを信用せず、軽蔑する。自己宣伝は、コミュニティがそこから何か得るものがあるように見えるときでさえ、容赦なく批判されることが多い。これがあまりに極端になっていて、ハッカー文化の「大物」や部族長老たちは自分の地位を保つためには話し方を穏やかにして、あらゆる機会をとらえてユーモアをこめて自己卑下することが要求されるようになってしまっている。この態度がいかにして、明らかにほぼ完全にエゴに依存して機能するインセンティブ構造とからみあっているのかは、どうしても説明が必要となるところだ。

もちろんその大きな部分は、欧米の「エゴ」に対する一般的に否定的な態度からきている。多くのハッカーの文化マトリックスは、エゴの満足を求めるのは悪い（あるいは少なくとも大人げない）と教えている。そしてエゴはせいぜいがプリマドンナのみに許される奇矯な代物で、精神的な病理の兆候であることすら多い、と教わっている。これを「仲間内の評判」や「自尊心」「プロ意識」「達成感」というふう昇華して、はじめてうけいれられるものになるんだ、と。

ぼくたちの文化的伝承のこの部分の不健全な根っこについては、全く別のエッセイを書くことだってできる。あるいは、ぼくたちが真に「無私」の動機を持っている（心理と行動面でのあらゆる証拠にもかかわらず）と信じ込むことによる、自己欺瞞の害についてだって別の文章が書けるだろう。書いてもいいんだが、ぼくがやるまでもなく、「愛他主義」を破壊して、はっきり意識されない自己利益に還元することについては（その他のいろんな欠点はさておき）フリードリッヒ・ヴィルヘルム・ニーチェとエイン・ランドが、きわめて立派な作業をすでに残している。

でもここでやってるのは道徳哲学でもなければ心理学でもないもので、エゴが悪いものだという信仰から生じるちょっとした害を検討するにとどめよう。その害とはつまり、そのおかげで多くの

ハッカーたちは自分の文化の社会力学を意識的に理解することが、感情的にむずかしくなってしまうという点だ。

でも、この線での検討はまだ終わったわけじゃない。目に見えてエゴだけからくる行動に対するタブーは、われわれを取り巻く文化にもあるけれど、それはハッカー（サブ）カルチャーでは極端に強化されていて、だからそれがハッカーたちにとっては、なにか一種の特殊な適応上の機能があるにちがいないとしか考えられない。その他多くの贈与文化、たとえば演劇人たちの同業者仲間や大金持ちの間では、このタブーはもっと弱いんだから！

11 謙虚さの美德

ハッカー文化の報酬メカニズムにおいて名声が中心的なものだということがはっきりしたので、今度はなぜこの事実が半ば隠されて、ほとんど公認されないのが重要と思われていたのかを理解しなけりゃならない。

海賊ソフト文化との対比は示唆的だ。この文化では、地位を求めての行動は臆面なしで、これ見よがしですらある。この手のクラッカーたちは、「ゼロ・デイ・warez」(クラックされていないオリジナルバージョンがリリースされたその日に、クラックされたソフトを配布すること)をリリースして評判を勝ち取ろうとするけれど、でもそのやり方については口をつぐむ。この種の魔法使いどもは自分の小技を公開するのを嫌う。だから結果として、クラッカー文化の知識ベースはごくゆっくりとしか成長しない。

ハッカーコミュニティでは、それとは対照的に、ある人の成果こそがその人の主張でもある。ここには非常に厳格な能力主義(いちばんすぐれた職人性が勝つ)があって、そして品質は自ら語るべきだ(いや、語らなくてはいけない)という倫理が強く存在している。いちばんすてきな自慢は、「とにかく動く」コードであり、そして有能なプログラマならだれでもこれがいいのがわかるだろうというものだ。だからハッカー文化の知識ベースは急速に拡大する。

エゴだけに基づく態度に対するタブーは、したがって生産性を増す。でも、これは副次的な効果にすぎない。ここで直接守られているのは、コミュニティの同業者評価システムの中での情報の品質だ。つまり、大仰な自慢や夜郎自大ぶりが抑圧されているのは、それが創造的、協調的な行動におけるだいたいな信号をゆがめてしまうノイズのように作用しがちだからだ。

これととてもよく似た理由のために、コードではなくその著者を攻撃するということには行われなない。ここには、小さいながら大事な点があって、ぼくの論点を補強してくれる。ハッカーたちは、イデオロギー上のちがいや個人的なちがいでおたがいをフレームしあうのは、ぜんぜんかまわないと思っている。でも、どんなハッカーでも、技術的な仕事について、ほかのハッカーの有能ぶりをおおっぴらに罵倒するのは、前例がない(私的な批判でさえ、あまり見られないし、あってもかなり口ごもった感じで行われる)。バグの追跡と批判は必ずプロジェクトの名前で行われ、個人の名前では行われなない。

さらに、開発者は過去のバグのことで自動的に責められたりはしない。バグが修正されたとい

うことのほうが、そのバグがかつて存在したということよりも重要だと一般には考えられている。ある評者が述べたように、「Emacs のバグ」を直すことで自分の評判をあげる人はいるけれど「リチャード・ストールマンのバグ」をなおして評判を上げる人はいない。そしてストールマンを批判するのに、修正済みの古いバグを根拠にするのは、とてもよくないこととされるだろう。

これは、アカデミズムの多くの部分ととてもおもしろい対照ぶりを示している。アカデミズムでは、欠陥があると思われる他人の成果をボロクソにするのは、評判を勝ち取る重要な方法の一つだ。ハッカー文化では、こうした行動はいささか強気にタブー視されている。それが強力すぎるために、そういう行動が存在しないということ、を、ぼく自身が思いつかなかったほどだ。ちょっと変わった視点を持つ評者が、本論文発表から1年以上たって指摘してくれて、やっとはっきり気がついたほどだ！

能力に対する攻撃に関するタブー（アカデミズムとは異なる）は、威張ることについての（アカデミズムと共有された）タブーよりさらに多くを物語ってくれる。ハッカー世界とアカデミズム世界での、コミュニケーションのあり方のちがいや支持構造のちがいについて考えるとき、これが足がかりになるからだ。

ハッカー文化の贈与メディアは実体がなく、そこで使われるコミュニケーションの回線は感情的なニュアンスを表現するには乏しい。メンバー間のフェイス・トゥ・フェイスの接触は、例外的な現象であるほうが多い。このため、ノイズに対する許容度はほかの贈与文化よりかなり低いし、これで能力に関する攻撃についてのタブー、かなりよく説明できるだろう。ハッカーたちの能力についてのフレーム事例は、この文化の評判スコアボードを許容できないくらい乱すものなんだ。

同じようなノイズに対する脆弱さは、ハッカーコミュニティの部族の長老たちに要求される、公的な場での謙虚さも説明してくれる。長老たちは、傲慢さや尊大さに冒されていないと見られなくてはならない。そうしないと、危険なノイズに対するタブーが維持されないからだ。[DC]

あたりが柔らかいのは、うまくいくプロジェクトの管理者になりたいときにも役に立つ。その人は、自分がよい判断力を持っているとコミュニティに納得させなくてはならない。管理者の仕事のほとんどは、ほかの人のコードを判断する作業だからだ。自分たちのコードの質が明らかに理解できないような人物に、自分の作業を貢献しようと思うやつはいない。あるいは、プロジェクトの評判という収益を不公平に着服しようとしそうな行動がうかがえるようなヤツに？ 潜在的な貢献者たちとしては、プロジェクトリーダーに十分な謙虚さと気高さを求める。そして客観的にみて正しいときにはこういえる人物であってほしい：「うん、たしかにそれは、ぼくのバージョンよりうまく動く。そっちを使おう」そしてその際には、しかるべきところにクレジットを与えてくれる人物であってほしいんだ。オープンソース界での謙虚な行動の理由としてもう一つあるのは、プロジェクトが「終わった」という印象をなるべく与えたくない、ということがある。そんな印象ができれば、潜在的な貢献者は、自分が必要とされていないような気分になるかもしれない。自分の力を最大化しておくには、プログラムの現状について慎ましい態度をとることだ。自慢はコード自身にさせておいて、口では「ああちくしょう、このソフトはまだ X も y も z もできない、まだぜんぜんダメだ」と言えば、x、y、z 用のパッチはすぐに出てくることが多い。

最後にぼくは個人的に、一部の代表的ハッカーたちの自己卑下行動は、個人崇拜の対象となることに対する本物の（そして正当でなくはない）おそれからきているのを観察してきた。リーヌ・トヴァルズとラリー・ウォールはどちらも、こうした忌避のはっきりした例をたくさん提供してくれる。まえにラリー・ウォールと夕食にでかけたとき、ぼくはこう冗談をたいた。「ここで最高のハッカーはきみだ　だからきみがレストランを選んでくれ」するとかれは、はっきりとたじろいだ。もっともな話だ。共有した価値と、リーダの価値観とを区別できなくなったことで、ダメになってしまったコミュニティは多い。このパターンについてかれもリーヌも思い知っているはずだ。一方で、多くのハッカーはラリーの問題を自分も抱えてみたいものだと思うだろう。それを自分で認める勇気があればだけれど。

12 評判ゲームモデルが持つグローバルな意義

評判ゲーム分析は、一見するとわかりにくいかもしれない含蓄をもっと持っている。その多くは、既存のプロジェクトに協力するより成功プロジェクトを創始するほうが名声が高まるという事実から派生するものだ。また、既存のプロジェクトをだんだん改善させるような、「おれもおれも」式改良よりは、感動するほど独創的なプロジェクトのほうがポイントを稼ぎやすい。一方で作者以外のだれにも理解できない、あるいは作者にしかニーズのないソフトは、評判ゲームでは何の役にもたたないし、みんなに新しいプロジェクトを認識してもらうよりは、既存プロジェクトに貢献したほうが、高いポイントを稼ぎやすい。最後に、空っぽのニッチを埋めるよりは、すでに成功しているプロジェクトと競合するほうがずっとむずかしい。

したがって、ご近所（いちばん近い競合プロジェクト）からの最適な距離ってものがある。近すぎればその成果物は「おれもおれも！」式の価値の低いものとなり、贈り物としては弱い（それなら既存プロジェクトに貢献する方がいい）。あまりに離れすぎていたら、だれもその努力を利用できず、理解できず、そして重要性を理解することもできなくなる（これまた贈り物としては弱い）。これは、ノウアスフィアの開墾パターンの面で、物理的なフロンティアに散らばって入植する入植者たちといささか似たような光景をつくりだす　ランダムではないが、散らばりかたの限られたフラクタルの波のようなパターンだ。プロジェクトは、フロンティア近くの機能的ギャップを埋めようとして創始される傾向が強い。（目新しさの誘惑に関する追加の議論については [NO] を見てほしい）

一部のすごく成功したプロジェクトは「カテゴリーキラー」となる。だれもその近くを開墾したいとは思わない。確立された基盤とはりあって、ハッカーたちの関心を求めるのはむずかしすぎるからだ。そうでなければ独自のプロジェクトを創始するような人たちは、かわりに大きな成功したプロジェクトに拡張を加えるにとどまる。古典的な「カテゴリーキラー」の例は GNU Emacs だ。その変種は、完全にプログラム可能なエディタの生態ニッチをあまりに完璧に埋めてしまったので、だれも 1980 年代初期以来、まったく異なるデザインなど試してみようとすらしらない。かわりにそういう人は、Emacs のモードを書くんだ。

全体的にみて、この二つの傾向（ギャップ埋めとカテゴリーキラー）のおかげで、時代を追うにつれてのプロジェクト開始傾向、おおむね予想がつく。1970年代に出回っていたほとんどのオープンソースはおもちゃとデモだった。1980年代には、開発ツールとインターネットツールが主流。そして1990年代になると、中心はOSに移った。いずれの場合にも、それ以前の分野の可能性がほとんど尽きてしまって、新しいもっと難しいレベルの問題に矛先が移っている。

このトレンドは、近未来にとっておもしろい意味を持っている。1998年頭には、Linuxはどうやら「フリーOS」というニッチにとってカテゴリーキラーとなったような感じだ。かつてなら競合OSを書いたかもしれない人たちは、いまではLinuxのデバイスドライバや拡張を書いている。そしてこの文化がこれまで想像してきたようなローレベルのツールは、ほとんどすべてオープンソースで存在している。あとはなにが残っているだろう？

アプリケーションだ。2000年が近づくにつれて、オープンソース開発がますます最後の処女地に向かってシフトすると予言してもよさそうだ。その処女地とはすなわち、技術オンチのためのプログラムだ。はっきりした最初のしるしとしては、GIMP (<http://www.gimp.org/>)の開発がある。これはフォトショップに似た画像処理ソフトで、過去10年には商業アプリケーションの *de regueur* と思われていた、エンドユーザにやさしいGUIを持っている。別の示唆としては、アプリケーション用ツールキットのプロジェクトを取り巻いているにぎやかさが指摘できるだろう。KDE (<http://www.kde.org/>) や GNOME (<http://www.gnome.org/>) なんかだ。

この論文の評者の一人は、開墾のアナロジーを使うことで、ハッカーたちがなぜマイクロソフトの「抱き込んで拡張」方針、つまりインターネットプロトコルを複雑化して閉じたものにしてしまおうという方針に対し、なぜあれほど露骨な怒りをもって反応するのか、ということが説明できることを指摘している。ハッカー文化はほとんどのクローズドなソフトと共存できる。たとえば Adobe Photoshop があっても、そのオープンソース版とも言うべき GIMP 周辺の領域が極端に魅力がなくなるわけではない。でも、マイクロソフトがプロトコルを「脱・共有化」[HD] してしまっただけで、マイクロソフト自身のプログラマしかそれを使うソフトが書けないようにしてしまったら、マイクロソフトは単に独占を拡大して顧客に害を与えるだけではすまない。ハッカーたちが開墾して耕作すべきノウアスフィアの量と質を低下させることにもなる。ハッカーたちがマイクロソフトの戦略を「プロトコル汚染」と呼ぶのも当然だろう。かれらは、自分の農業用水に毒を入れようとしている人を見つけた農夫たちとまったく同じ反応を示しているわけだ！

最後に、評判ゲーム分析はよく挙げられる格言を説明してくれる。人はハッカーを名乗ればハッカーになれるわけじゃない。ほかのハッカーにハッカーと呼ばれた時点で、人はハッカーになるんだ。この観点から見た「ハッカー」というのは、技術的な能力を持っていて、しかも評判ゲームがどう機能するかわかっているということ（贈り物を提供することで）実証した人間だ。この判断は、おもに認知と文化順応に基づくもので、すでにこの文化の中にしっかり入った人々からしか与えられない。

13 どれほどすばらしい贈り物？

ハッカー文化が貢献を評価して、それに対する仲間内の評判を戻すやり方には、一貫したパターンがある。以下のような規則はすぐに見分けられる：

1. 期待させられたほどきちんと機能しないなら、それはダメ　　どんなに巧妙でオリジナリティ豊かなものであっても。

「期待させられたほど」という部分に注意。このルールは、完全無欠を求めるものではない。ベータ版や実験ソフトには、もちろんバグがあってもいい。これは、プロジェクトの段階や開発者の主張をもとに、リスクを正確に評価できるようにしろ、と求めているわけだ。

このルールのおかげで、オープンソースソフトはずいぶん長い間ベータ版のままになっていて、とんでもないバグがぼろぼろ出てきたりしない、というのを開発者が革新するまで、バージョン 1.0 を名乗ったりしない、という現象が見られる。クローズドソースの世界では、バージョン 1.0 というのは、「慎重な人は手を出しちゃダメよ」という意味だ。オープンソースの世界では、むしろ「開発者たちはこいつに自分の評判を賭けてもいいと思っています」というのに近い。

2. 既存の機能する領域を複製する作業よりは、ノウアスフィアを拡張する仕事のほうがいい。

こいつをもっと素朴に表現するなら、「独創的な作業は、既存ソフトの機能を複製するよりもいい」ということになるだろう。でも、実はこんなに単純ではない。既存のクローズドなソフトの機能を複製するのは、オリジナルなソフトの開発作業と同じくらい高い評価が得られる　それによって、クローズドなプロトコルやフォーマットをこじあけたりして、その領土を新たに提供できるのであれば。

だからたとえば、いまのオープンソース世界でいちばんプレステージの高いプロジェクトは、Samba だ　　Unix マシンがマイクロソフトの独占ファイル共有プロトコル SMB のクライアントまたはサーバとして使えるようにするコード。創造的な仕事はほとんどない。むしろ細かいリバースエンジニアリングの細部をうまくやるのが一番の問題だ。それでも、Samba グループは英雄視されている。ユーザ集団をまるごと困り込んで、ノウアスフィアの大きな部分に縄を張ってしまおうというマイクロソフトの努力を無化するものだからだ。

3. 主要ディストリビューションに含まれるものは、そうでないものよりもいい。主要ディストリビューションすべてに含まれるものがいちばんえらい。

ここでの主要ディストリビューションというのは、Red Hat、Debian、Caldera、S.u.S.E. など Linux の大手ディストリビューションのみならず、独自の評判を持っていて、したがって品質を裏書きできるとされているその他のコレクションすべてを含んでいる　たとえば BSD ディストリビューションや、FSF のソースコレクションなど。

4. 利用は最大の賛辞　　そしてカテゴリーキラーは追従ソフトよりもよい。

他人の判断を信用するというのは、同業者（ピア）レビューのプロセスの基本だ。これが必要なのは、だれもあらゆる代替ソフトをすべてレビューするだけの暇はないからだ。だから、多くの人が使っているソフトは、少数の人しか使っていないソフトよりもいいとされる。

あまりにできがよくて、ほかの代替品をだれも使おうとしなくなるような仕事をするというのは、だからすさまじいプレステージを獲得することになる。最大の同業者評価を獲得するには、非常に人気のあるカテゴリキラーとなった独創的な作業が、すべての主要ディストリビューションに含まれるようになることだ。これを一度ならず行った人間は、半ば冗談まじりで「半神半人」(demigods) と呼ばれる。

5. むずかしくて退屈な作業 (たとえばデバッグやドキュメンテーション書き) に継続的に貢献するのは、おもしろくて簡単なハックをいいとこ取りするよりも賞賛すべきことである。

この規範によって、ハッカーたちが通常はやりたがらないけれど必要な仕事について、コミュニティは見返りを支払うわけだ。でもこれは、次のルールとはちょっと矛盾している：

6. 機能の大幅な (non-trivial な) 拡張は、低レベルのパッチやデバッグよりもいい。

これがどう機能するかというと、こういうことらしい：一発だけで見た場合、機能を追加するほうがバグ取りよりも見返りが大きい。ただしバグが極端にてごわかったり見つけにくかったりして、それを仕留めること自体が非凡なスキルや賢さを示すのでない限り。でも、こうした行動が長い期間にまたがると、長いこと注意を払って、ごくつまらないバグでもつぶしていく人は、同じ努力をかけて簡単な機能の追加をしてきた人よりも高い地位につけるわけだ。

ある評者が指摘したことだけれど、こうしたルールはおもしろい形で相互作用して、いつの時点でも最大の効用が報われるようには必ずしもなっていない。ハッカーに、独自の新しいツールを開発するのと、だれか別の人のツールの拡張と、どちらで有名になる見込みが高いかときいたら、答えはまちがいでなく「新しいツール」になるだろう。でも、次の二つならどうだろう。

1. 一日ほんの数回、だれにもわからない形で OS から呼び出されるだけの真新しいツールだけれど、急速にカテゴリキラーになるもの
2. ことさら新しくもないしカテゴリキラーでもない既存ツールに対する拡張をいくつか。ただしそのツールは、毎日使われて、毎日多くの人の目に直接触れる

この場合、ハッカーは最終的に前者に落ち着くにしても、かなりためらうはずだ。この両者は、だいたい同じくらいの地位を持っている。

これを指摘した評者は、以下のような一言を付け加えて、ぼくに対して論点を強調した。「前者の例は、fetchmail で、後者の例はあなたの場合、vc.el や gud.el など各種の Emacs 拡張です」たしかにかれの言うとおり。ぼくは「各種の Emacs モードの著者」と言われるよりは、「fetchmail 作者」と言われるほうが多い。長期的に見れば、Emacs モードのほうが効用の合計では大きいにちがいないのだけれど。

ここで起きているのは、真新しい「ブランドアイデンティティ」を持つ仕事は、既存「ブランド」に集積する仕事よりも認知度が高い、ということにすぎないのかもしれない。こうしたルールのさらなる解明と、それがハッカー文化の得点集計システムについてどんなことを物語るのかは、今後の研究課題として有望だろう。

14 ノウアスフィアの所有権となわばりの動物行動学

土地所有権の慣習がなにをもたらずか理解するには、もう一つ別の角度から見てみると役にたつ。動物行動学、とくになわばりの行動学だ。

所有地は、動物のなわばりの抽象化で、なわばりは種の内部の暴力を減らすための手段としてうまれた。境界をしるし、ほかの仲間の境界を尊重することで、オオカミは争いを始める機会を減らす。そういう争いで自分がけがをしたり殺されたりするかもしれない、そうなったら自分が再生産面で成功する可能性が低下するからだ。

同じように、人間社会における「所有物」の概念は、平和な行動と抗争とをはっきりわかるような境界をもうけることで、人間同士の紛争を予防するのが機能だ。人間の所有物なんて、ただの恣意的な社会的因襲にすぎないと表現するのはファッショナブルだったりするけれど、でもこれはとことんまちがっている。犬を飼ったことがある人で、知らない人が飼い主の所有物に近づいたときに犬が吠えたのを経験した人は多いだろう。そういう人ならみんな、動物のなわばり性と人間の所有物との本質的な連続性を経験しているはずだ。ぼくたちが家畜化した狼のいとこたちは、この点について人間の政治理論家たちの多くより本能的に賢いんだ。

所有権を主張すること（なわばりをマーキングするように）は、それを演じてみせることが重要で、その境界を守るぞという宣言だ。所有権主張に対するコミュニティの支援は、摩擦をへらして協利行動を最大化するための方法だ。これは「所有権主張」が README ファイルのプロジェクト管理者の名前記述のように、柵や吠える犬よりずっと抽象的な場合でも成立する。それはやっぱりなわばりを抽象化したもので、（その他所有権の形態と同じく）ぼくたちの本能に基づく所有権モデルは、なわばりが発達して紛争解決に役立つようになったものなんだ。

この論点は、ハイパーリンクや高性能サーチエンジンの存在で一層強化される。Web ページを持つプロジェクトは、ノウアスフィアで近所を探検している人にずっと認知されやすい。リンクする人もいるだろうし、サーチエンジンにもひっかかる。Web ページは、だから広告としても優れているし、パフォーマンス行為としても有効だし、領土主張としても強力だ。

この動物行動学的な分析な、一見するとすごく抽象的で、実際のハッカー行動と結びつけるのは難しそうだ。でもこれはすごく大事な意味を持っている。一つは World Wide Web のサイトの普及ぶりを説明することだ。特に、なぜ web サイトを持ったオープンソース・プロジェクトが、ないものよりずっと「本物」らしくて重要に見えるのか、という点。

客観的に見れば、これは説明がむずかしい。ほんの小さなプログラムであっても、それを創始して管理するための努力に比べれば、web ページなんかばかみたいに簡単だ。だから web ページが内容の高さや非凡な努力の証拠だとは考えにくい。

Web そのものの機能的な特徴だけでも、説明としては不十分だ。web ページのコミュニケーション機能は、FTP とメーリングリストと Usenet ポストとの組み合わせでも同じくらい、あるいはもっとうまく提供できるものだ。実際問題として、プロジェクトの日々の連絡が、メーリングリストや Usenet のポストではなく Web で行われるというのはあまりレイがない。それなら、なぜ

プロジェクトのホームとして web サイトに人気があるんだろう。

「ホームページ」という用語に含まれたメタファーがだいじなヒントを提供してくれる。オープンソースプロジェクトの創始は、ノウアスフィアにおける領土を宣言することだけれど（そしてふつうはそのように認識されてはいるけれど）、でも心理的なレベルからいえば、それだけではあまり説得力はない。ソフトウェアは結局のところ自然の場所もないし、すぐに複製可能だからだ。「領土」や「所有物」という本能的な考えに適合しないわけではないけれど、でもそれにはしかなるべき努力が必要になる。

プロジェクトのホームページは、比較的空間的に組織された World Wide Web の世界で「故郷」領土として宣言を行うことにより、可能なプログラムの空間の抽象的な開墾を具体化するものなんだ。ノウアスフィアから「サイバースペース」にありてきても、柵や吠える番犬の現実世界まではまだ距離があるけれど、でも抽象的な領土主張をもっと直感的ななわばり理解に結びつけてはくれる。だからこそ、Web ページを持ったプロジェクトはもっと「本物」に感じられるんだ。

この動物行動学的な分析は、オープンソース文化での紛争処理メカニズムをもっと詳しく見るよう示唆するものでもある。これが正しければ、所有権慣習は評判インセンティブを最大化するだけでなく、紛争の防止と解決にもなんらかの役割を果たしていると考えていいはずだからだ。

15 紛争の原因

オープンソース・ソフトをめぐる紛争の理由としては、次の4つの大きな問題を指摘できる：

- だれがプロジェクトについて決定力のある決断をくださのか？
- だれがなにについて、評価や非難を受けるのか？
- 二度手間をさけて、非公式版がでてバグ追跡をややこしくするのを止めるにはどうするか？
- 技術的に見たら、なにが正しいことだったのか？

でもこの「技術的な正しさ」という点についてよく考えてみると、これは問題にはならない。この手の問題には必ず、万人が納得する客観的な決定方法があるか、あるいはないかのどちらかだからだ。もしそれがあれば、話はそこでおしまいでみんな勝つ。もしないのなら、それは「だれが決めるんだ？」という話に還元される。

同じように、紛争解決理論がプロジェクトについて解決しなくてはならない三つの問題は (A) 設計上の決断を下すときに最終的な采配を下すのはだれか、(B) どの貢献者がどのようにクレジットされるかをどう決めるか、(C) プロジェクトグループやその成果物が複数に枝分かれして分裂するのをどう防ぐか、となる。

問題 (A) と (C) の解決のために所有権慣習の果たす役割ははっきりしている。習慣により、プロジェクトの所有者こそが方向を左右する決定を下すことが認められる。そしてプロジェクト分裂による勢力の希薄化に対しては、慣習が強い圧力をかけることはこれまで見た通り。

ここで、評判ゲームは忘れてこれをハッカー文化についての純粋な「職人氣質」モデルから見た場合にも、こうした慣習は筋が通っていることは認識しておくといいたいだろう。この見方では、こうした慣習は評判インセンティブの希薄化よりはむしろ、職人が自分のビジョンを自分の選ぶ形で実行するという職人の権利を守ることを重視していることになる。職人モデルは、問題(B)「だれがなにについてクレジットを与えられるか」についてのハッカー慣習を説明するには十分ではない(だって、純粋な職人は評判ゲームなんか気にしないから、そんなことを気にする動機がない)。これを分析するには、ロック理論をもう一步進めて、所有権上の抗争や運用がどう行われているかをプロジェクト同士だけでなく、プロジェクト内部から見てやらなくてはならない。

16 プロジェクト構造と所有権

簡単な例は、プロジェクトの所有者・管理者が一人しかいない場合だ。この場合、紛争は起こり得ない。所有者がすべての決定をくだして、賞賛も批判もすべて自分で背負う。紛争として唯一あり得るのは引き継ぎ問題だけだ。もとの所有者がいなくなったり興味をなくしたりした場合、だれが新しい所有者になるのか。コミュニティとしても、問題(C)から、プロジェクトの分裂を避けたいと考える。こうした利害は文化規範として、所有者・管理者は自分がプロジェクトを管理できなくなった場合には、公式にその権利をだれかに譲るべきだとかたちで表現される。

いちばん単純ながら、簡単ではないケースとしては、プロジェクトを所有する一人の「優しい独裁者」のもとに複数の共同管理者が働いているような場合だ。慣習は、グループプロジェクトについてはこのモードを好む。これは Linux カーネルや Emacs ほどの大プロジェクトでも使えるし、「だれが決める」問題も解決してくれるし、その解決法も、その他のあらゆる代替案より特にひどいというわけでもない。

典型的には、優しい独裁者組織は、創始者が貢献者を集めるにしたがって所有管理者組織から発展して生まれる。所有者が独裁者であり続けても、それはプロジェクトのどの部分についてだれがクレジットを得るかという点について新しいレベルの紛争の可能性を持ち込むことになる。

この状況では、慣習は所有者/独裁者が公平に貢献者にクレジットを与えることを義務づける(たとえば README 履歴ファイルの中でのしかるべき記載なんかで)。ロック式所有権モデルでいえば、これはつまりプロジェクトへの貢献によってその評判リターンの一部を稼げるということだ(いいものも悪いものも)。

この論理をつきつめると、「優しい独裁者」はその全プロジェクトを文句なしに占有しているわけじゃないのがわかる。方向性を決める決断をする権利を持ってはいるけれど、かれは実際問題として、全体の評判の一部を他人の作業と交換にゆずりわたす。農場での分益小作人との対比はほとんど抵抗しがたいものがある。ただし、ハッカーの場合、貢献者の名前はクレジットに残り、その貢献者がもはや活発でなくなったときにも、ある程度の「収益」をあげ続ける点はちがうけれど。

優しい独裁者プロジェクトの参加者が増えるにつれて、参加者は二派にわかれる傾向がある。通常の貢献者と共同開発者だ。共同開発者となる典型的な道としては、プロジェクトの主要サブシス

テムの責任を負うことだ。もう一つは「高等修理脚」の役割を負って、多くのバグを抽出してなおすことだ。こういった方法を通じて、共同開発者は貢献者のなかでもプロジェクトに相当量の時間を投資し続ける。

サブシステム所有者の役割は、本論の分析ではときに大事なので、もっと詳しくみるべきだ。ハッカーたちは「権威は責任に伴う」というのが好きだ。あるサブシステムのメンテナンス責任を負う共同開発者は、そのサブシステムの実装と、その他プロジェクトとのインターフェース部分との両方をコントロールできるようになる。それを修正できるのは、アーキテクトとして働くプロジェクトリーダーだけ。このルールは、実質的にはプロジェクトのロック・モデル内部に、囲い込まれた所有地をつくりだしているものとみることができる。だからほかの領地境界とまったく同じように、紛争防止の役割を持つことになるわけだ。

習慣的に、共同開発者のいるプロジェクトの「独裁者」またはプロジェクトリーダーは、その共同開発者たちにだいじな判断のときには相談することが期待されている。これは特に、その決断が共同開発者「所有」のサブシステム（つまり、その共同開発者がそれに時間を注いで、責任を負っているサブシステム）にかかわるものであるときにあてはまる。賢いリーダーは、プロジェクト内部の領土境界の機能がわかっているから、サブシステム所有者の決定をじゃましたり、ひっくりかえしたりするようなことは軽々しくしない。

一部のすごく大きなプロジェクトは、「やさしい独裁者」モデルを完全に捨てている。これをやる方法の一つは、共同開発者たちを議決委員会にしてしまうことだ（Apache がこうしている）。別の方法としては、独裁権を巡回方式にすることだ。ここではコントロール権は、上級共同開発者たちのサークルのなかで次から次へとまわされる（Perl 開発者たちはこういう形の組織を採用している）。

こういう複雑な仕組みは、一般には不安定でむずかしいものと思われている。明らかにこのむずかしいという印象は、委員会による設計や委員会そのものの危険性の関数である面が大きい。こういう問題について、ハッカー文化ははっきりわかっている。でも、ハッカーたちが委員会や巡回式組織について生理的に感じている気持ち悪さは、それがもっと簡単な場合についてハッカーたちが理由づけるときに使う、無意識のロック式モデルにおさまりにくいことがあるんだろう。こういう複雑な組織では、コントロールという意味での所有権や、評判の収益の所有権をきちんとつけるのがむずかしいからだ。内部の境界がどこにあるかは見えにくいので、グループとして例外的に高い調和と信頼関係ができていない限り、紛争を避けるのはむずかしい。

17 紛争とその解決

これまで見てきたのは、プロジェクト内部では役割がますます複雑化し、それがデザイン決定権の配分と部分的な所有権によって表現されるということだった。これはインセンティブを配分する効率のよい方法だけれど、プロジェクトリーダーの権威を弱めるものでもある。いちばん大事な点として、リーダーが紛争をおさえこむための権威を弱めてしまうんだ。

設計にかかわる技術的な論争こそが、一見すると血を見るような争いにいたる一番の原因になりそうだと思うかもしれないけれど、でもこれは深刻なもめごとの原因にはめったにならない。こういうのは、権威は責任に伴うというなわばりのルールによってそこそこ簡単に解決がつく。

紛争解決手段としてもう一つあるのは、序列によるものだ。もし貢献者二人、または貢献者集団二つが争いを始めて、それが客観的に解決できず、どちらもその争いのなわばりを所有はしていないとき、そのプロジェクト全体につぎ込んだ作業量の多いほう（つまり、プロジェクト全体の中で地役権を最大に持つ側）が勝つ。

（同じように、投入したものが少ない方が負ける。おもしろいことに、これは多くのリレーショナルデータベース・エンジンが、デッドロックを解決するヒューリスティックとまったく同じだ。二つのスレッドが、あるリソースをめぐるデッドロックになったら、現在のトランザクションに投入したものがいちばん少ないほうが、デッドロックの犠牲者として選ばれて、終了させられる。これはつまり通常は、いちばん長く走っているトランザクション、あるいはいちばん上位のトランザクションが、勝者となるということだ。）

こういうルールだけで、ほとんどのプロジェクト上の争いを解決するには十分だ。十分でない場合にも、リーダーの采配でだいたいなんとかなる。この二重のフィルタを越えるほどの紛争はほとんどない。

紛争は原則的に、こうした二つの基準（「権威は責任にともなう」「序列の高いほうが勝つ」）が矛盾する結果を出して、しかもそのプロジェクトリーダーの権利が弱かったり不在だったりするときにしか生き延びない。これが起きるいちばんはっきりした例は、プロジェクトのリーダーが消えたあとで、その後継をめぐる争いだ。ぼくは一度、この種の争いに加わっていたことがある。みにくくて苦痛で、長引いて、最終的に解決されたのも、関係者がみんな疲れ切って、コントロールを外部の人に任せようと思ったから可能になった。もう二度のあの種のものには近づかずにすみますように、と心から願わずにはいられない。

最終的には、こういう紛争解決メカニズムは、ハッカーコミュニティがそれを強制しようという意志をもっているかどうかにかかってくる。そして唯一の強制メカニズムは、フレーミングと黙殺だ。慣習を破る者たちに対する公開糾弾と、破った者たちとはもう協力しないということだ。

18 文化への順応過程とアカデミズムとの関連

この論文の初期バージョンは、今後の研究テーマとして次の課題を挙げていた：ハッカー・コミュニティはそのメンバーに、慣習をどうやって伝えて指導するのか？ 慣習は自明なものなのか、あるいは無意識のレベルで自己組織的なものなのか、事例によって教えられるのか、それとも明文化された指示によって教えられるのか？

明文化された指示による教育は明らかにほとんどない。そもそもこの文化の規範についての明文化された記述は、いままでほとんどなかったから使いようがなかった。

多くの規範は事例によって教えられる。すごく簡単な例を挙げるなら、あらゆるソフトの配布

パッケージは README や READ.ME というファイルを含むべきだという規範がある。この慣習は 1980 年代初期には確立されていたものだけけれど、でもいまのところそれが明文化されたことはない。みんな、いろんなパッケージを見た結果としてその規範を知るわけだ。

一方で、ハッカー慣習のなかには、評判ゲームについての基本的な（おそらくは無意識の）理解さえ獲得できたら自己組織的に導かれるものもある。第 3 章でぼくが挙げた 3 つのタブーについて、ハッカーのほとんどはわざわざ教わる必要はないし、そうでなくても、あんなのは教えるまでもなく自明のことだというだろう。この現象はもっと詳細な分析が必要だ。そしてその説明は、ハッカーがハッカー文化についての知識をどうやって得るかというプロセスを見ればわかるかもしれない。

文化順応メカニズムとして、隠されたヒント（宗教・神秘学的な意味ではもっと厳密な「謎」）を使う文化は多い。これは部外者には空かされない謎だが、熱心なタコ（newbie）なら見つけだしたり演繹したりできると期待されている。内部に受け入れられるためには、その謎が理解できていて、しかもそれを文化的に認められた方法で学んだことを示さなくてはならない。

ハッカー文化は、こういうヒントやテストを異常なほど意識的かつ広く利用している。このプロセスは、少なくとも三つのレベルで機能していることがわかる。

パスワード風の特定の謎。一例として、USENET には alt.sysadmin.recovery というニュースグループがあって、こういう秘密の例として非常にはっきりしたものを持っている。その秘密を知らないとメッセージをポストできない。そしてその秘密を知っているということが、ポストするのにふさわしい人物だという証拠と見なされる。常連はこの秘密をばらすことについて、強いタブーを持っている。

ある技術的な謎へのイニシエーションが必要なこと。価値ある贈り物ができるようになるためには、人は相当量の技術的な知識を吸収しなくてはならない（たとえば、主要コンピュータ言語のどれか一つを知らなくてはならない）。この必要条件が大きな場で果たす役割は、小さな集団で隠されたヒントが果たす役割と同じで、この文化の中で活動するのに必要な資質（たとえば抽象思考能力、根気、精神的な柔軟性など）のフィルタとして機能する。

社会的な文脈の謎。人がハッカー文化に関わりを持つのは、特定のプロジェクトへの参加を通じてだ。各プロジェクトはハッカーたちの生々しい社会的な文脈であり、そこへの参加者になりたい人間が活動するためには、いろいろ調べて技術的な理解とともに社会的な理解を身につけなくてはならない（具体的には、これをやる一般的な方法はそのプロジェクトの Web ページを読んだり、e-mail アーカイブを読んだりすることだ）。こういうプロジェクト集団を通じて、タコ（newbie）たちは経験豊かなハッカーたちの行動事例を経験する。

こういう謎を身につける過程で、ハッカー候補は文脈的な知識を身につけてゆき、（しばらくすれば）そのおかげで確かに三つのタブーやその他の慣習は「あたりまえ」に思えてくるわけだ。

ついでながら、ハッカー贈与文化の構造そのものが、それ自体中心的な謎だと論じる人もいるかもしれない。評判ゲームやそこにこめられた慣習、タブー、利用法について、本能的な理解を実証してみせない限り、文化に順応したとはみなされない（具体的にはだれもその人をハッカーとは呼ばない）。でもこれは些末なことだ。あらゆる文化は、参加者候補に対してそういう理解を要求する。さらにハッカー文化は、内的論理や民俗的な方法を秘密にしておこうなんていう希望はぜんぜん持っていない。少なくとも、それをばらしたぼくに対してフレームしてきた人はだれもいない！

名前を挙げきれないくらい多数のコメントが寄せられ、ハッカーの所有権慣習は学問世界、特に科学研究コミュニティの行いと密接に関連し、そこから直接派生したものかもしれないと指摘してくれた。こうした研究コミュニティは、潜在的に生産的なアイデアのなわばりを掘り起こすという面で似たような問題を抱えているし、同業者のレビューや評判を使うという点で、その問題に対して非常に類似した適応解決手段を示している。

多くのハッカーたちは、アカデミズムの世界と正式な接触を持ったことがある（大学でハッキングを学ぶのはよくあることだ）。だからアカデミズムの世界が適応パターン面でハッカー文化とどれほど共通性があるかという話は、こういう慣習の適用方法を理解するにあたっては、ついで以上の興味を向けるべき課題だ。

ぼくが特徴づけたようなハッカー「贈与文化」との明らかな類比が、アカデミズムの世界でもたくさん見られる。研究者が終身職を手に入れると同時に、もはや生存の問題は木にする必要がなくなる（実は終身職の概念は、「自然哲学者」たちがおもに金持ちの紳士たちで、研究に没頭するだけの暇をもっていた、初期の贈与文化までさかのぼることができる）。生存問題がない以上、評判の向上が作業をつき動かす目標となり、それは雑誌などのメディアを通じて新しいアイデアと研究の共有を奨励する。これは客観的にみて、機能的に筋が通っている。科学研究はハッカー文化と同じく、「巨人の肩に立つ（先人の成果に積み重ねる）」という考え方に大きく依存しているからだ。これにより、同じ基本原理をなんども発表しなおさずにすむ。

ある人はこの議論を極端にすすめて、ハッカー慣習は単に研究コミュニティの民俗的な慣習の反映にすぎず、実は（ほとんどが）そこで獲得されたものだ、とまで主張する。これはたぶん話を誇張させすぎているだろう。ハッカー慣習は頭のいい高校生でもすぐに獲得できてしまうという点だけでもそれは明らかだと思う。

ここにはもっとおもしろい可能性がある。ぼくの憶測だけれど、学問の世界とハッカー文化が同じ適応パターンを示すのは、それが出自の点で親戚だからではなく、物理法則と人間の本能の仕組みを千手としたときに、それぞれ自分たちがやろうとしていたことを実現するための、唯一最適な社会組織を発達させたんじゃないだろうか。歴史的にみて、自由市場資本主義こそが経済効率を求めて協力するための世界的に最適な方法だという審判はくだったようだ。ひょっとして同じように、評判ゲームに基づく贈与文化は、高品質の創造的作業を産み出し（そしてチェックする！）ために協力するにあたって、世界的に最適な方法なのかもしれない。

この点は、もし本当なら、ただの（失礼）学問的興味以上のものとなる。それは『伽藍とバザー

ル』³での考察を、ちょっと別の角度からも論証しなおしたものとなる。つまり多くのプログラマがポスト稀少性時代の贈与文化の中で生きられるだけの富の余剰を資本主義が作り出せるようになった瞬間から、最終的にはソフト生産の産業資本主義様式は競争に負けるべく運命づけられているのだ、という考察だ。

19 結論：慣習から慣習法へ

これまでオープンソース・ソフトの所有権とコントロールを支配する慣習をみてきた。その根底にある理論が、ロック式の土地所有権理論と似た所有権概念であることもみた。それと結びつけて、参加者が時間とエネルギーと創造性をあげてしまうことで名声を競う「贈与文化」としてハッカー文化を分析してみた。そしてその分析が、文化内の紛争解決にとって持つ意味について検討してみた。

論理的に考えて次にたずねるべき質問は「それがどうしたの?」というものだ。ハッカーは意識的な分析なしにこうした慣習を発展させてきたし、(いままでは)意識的な分析なしにそれにしがって来た。それを意識的に分析してみたところで、なにか実用性のある成果が得られたかどうかは、すぐにははっきりしない。ただし、これによって記述から処方へと歩みを進めて、こういう慣習の機能を改良する方法が抽出できるなら話は別だろう。

英米慣習法の伝統の土地所有権理論が、ハッカー慣習と論理的にとっても近い関係にあることを見つけた。歴史的にみて [Miller]、これを発明したヨーロッパの部族文化は、明文化もはっきり意識化もされない慣習から部族の賢者たちが記憶するはっきりした慣習法体系に移行し、やがてはそれを成文化することで、紛争解決システムを向上させていった。たぶん、人数がふえてきて新メンバー全員の文化順応を行うのがだんだん難しくなってくるにつれて、ハッカー文化もなにかそれと似たようなことをする時期がきているんだろう。オープンソース・プロジェクトとのかかわりで生じる可能性のあるさまざまな紛争を解決するためのよい方法を、明文化したコードとして開発し、コミュニティの長老メンバーが紛争調停を行うような、仲裁の伝統をつくりあげることだ。

この論文での分析から、これまで内在的だったものを明文化するようなそういうコードの概略が、どんなものになるべきかが示唆される。こういうコードは決して上から押しつけられるものではあり得ない。個別プロジェクトの創始者や所有者が、自主的に採用するものでなくてはならない。そしてそれは完全に硬直したものでありえない。文化にかかる圧力はだんだん変わるはずだからだ。最後に、こういうコードが実行上うまく機能するには、ハッカー部族の幅広い合意を反映したものでなくてはならない。

ぼくはそういうコードを書く作業を始めていて、ぼくが住んでいる小さな町の名前をとってそれを仮に「マルヴァーン綱領」と名付けている。もしこの論文の一般的な分析がそれなりに広く受け入れられたら、マルヴァーン綱領を紛争解決コードの見本として公開する。このコードの批評や開発に興味のある向きや、単にそれがいいか悪いかフィードバックをくれたい人は、電子メールでぼ

³邦訳:<http://cruel.org/freeware/cathedral.html>、原文:<http://www.tuxedo.org/esr/writings/cathedral-bazaar/>

くにコンタクトしてほしい (mailto:esr@thyrsus.com)。

20 これからの研究の課題

ハッカー文化 (そしてぼく) は、優しい独裁者モデルに従わない大規模プロジェクトについてあまりよく理解できていない。そういうプロジェクトの多くは失敗する。でも少数はものすごい大成功をおさめて重要になる (Perl, Apache, KDE)。ちがいがどこにあるのか、だれにもわかっていない。(そういうプロジェクトはそれぞれ *sui generis* (まったく独自のもの) であり、たまたまそのメンバーのグループ力学に基づいて成功したりしなかったりするのだというのが、漠然とただよう理解だけれど、でもこれって本当なんだろうか、それともある集団がとるべき複製可能な戦略があるのだろうか?)

21 書誌、注

[Miller] Miller, William Ian; *Bloodtaking and Peacemaking: Feud, Law, and Society in Saga Iceland.*; University of Chicago Press 1990, ISBN 0-226-52680-1. アイスランドの民会法についてのすばらしい研究。ロック式所有権理論がどれほど古いかを明らかにするとともに、慣習が慣習法へとひきつがれ、そして成文法になる歴史的なプロセスの後半を描いている。

[Mal] Malaclypse the Younger; *Principia Discordia, or How I Found Goddess and What I Did To Her When I Found Her.*; Loompanics, ISBN 1-55950-040-9. すごく軽い冗談口でんこもりだけれど、そのなかの「SNAFU 原理」なる代物は、上意下達方式がなぜスケラビリティが低いのかについて、なかなか深遠な分析を提供している。ブラウザ可能な HTML 版あり: (<http://www.cs.cmu.edu/~tilt/principia/>)

[BCT] J. Barkow, L. Cosmides, and J. Tooby (Eds.); *The Adapted Mind: Evolutionary Psychology and the Generation of Culture.* New York: Oxford University Press 1992. 進化心理学へのすぐれた入門書。論文のいくつかは、ぼくが論じている 3 種類の文化タイプ (上意下達、交換、贈与) と直接かかわるもので、こういうパターンが人間の心理のかなり深いところにまで食い込んでいるらしいことを示唆している。

[MHG] Goldhaber, Michael K.; *The Attention Economy and the Net.*

(http://www.firstmonday.dk/issues/issue2_4/goldhaber)

この論文はバージョン 1.7 のあとで発見した。一目でわかる欠点もある (Goldhaber は、関心というものには経済的な理由づけが適用できないと論じているけれど、この議論はよく読むと成り立たない) けれど、それでも Goldhaber は組織行動における関心集めの役割につ

いておもしろくて鋭いことを言っている。ぼくが論じた名声や仲間内の評判は、かれが言う意味での関心の特殊例として見ると有益。

[HH] ハッカー社会 (hackerdom) の歴史については、ハッカー界小史⁴にまとめてみた。これをほんとうにきちんと説明してくれる本は未だに書かれていないし、それを書くのはたぶんぼくではないと思う。

[N] 「ノウアスフィア (noosphere)」は哲学特有の専門用語で、「ノウ・ア・スフィア」が正しい発音⁵。表記について死ぬほど厳密でありたいなら、正しくは二番目の o の上に点々がつく。もっと詳しく言えば：この「人間の思考の圏域」を表す用語は、ギリシャ語の「nous」、つまり精神や意志、息などを指す単語から生まれている。E. LeRoy が *Les origines humaines et l'évolution de l'intelligence* (Paris 1928) で発明したことばだ。これはさらに、ロシアの生物学者にして先駆的なエコロジスト Vladimir Ivanovich Vernadsky (1863-1945) によって広められ、さらにはイエズス会の古生物学者兼哲学者ピエール・ティヤール・ド・シャルダン (Pierre Teilhard de Chardin, 1881-1955) がこれを広めた。いまではこの用語は主に、人間が将来純粋な精神体へと進化して神様と融合するというかれの理論と結びついて考えられている。

[SP] Linux と BSD の世界とのおもしろいちがいの一つは、Linux のカーネル (および関連する OS の中心的なユーティリティ) は一度も分裂していないけれど、BSD は少なくとも 3 回は分裂しているということだ。これがおもしろいのは、BSD グループの社会構造は中央集権化されていて、権限をはっきりと線引きすることで分裂を防ぐようになっているのに、中央集権化されていなくて不定型な Linux コミュニティは分裂を防ぐような手段をまったく講じていないからだ。開発をなるべく開放したほうが、いちばん分裂しにくくなると思えない！

Henry Spencer (henry@spsystems.net) の考えによれば、一般に政治的なシステムの安定性は、その政治プロセスへの参入障壁の高さに反比例するのではないかとのこと。かれの分析は、ここで引用するだけの価値がある：

「比較的オープンな民主主義の大きな強みの一つは、ほとんどの潜在的革命家は自分の目的を達成するのに、システムを利用したほうがシステムを攻撃するよりも簡単だと判断する、ということだ。各種組織が協力して「敷居を上げて」、不満を持った小集団が自分の目標を多少なりとも達成するのをむずかしくすると、この強みはすぐに消えてしまう。

(同じような原理は経済でも見られる。開放市場では競合がいちばん激しく、さらに最高で最低価格の製品が得られるのがふつうだ。このため、既存企業とし

⁴<http://earthspace.net/esr/faqs/hacker-hist.html>, 邦訳は <http://cruel.org/freeware/hackerdom.html>

⁵訳注：日本ではヌースフィアとかヌースフェルとか書かれることが多い。この翻訳では著者にしたがった。

では、新規参入をなるべくむずかしくするのがいちばん利益になる。たとえば政府に対し、コンピュータに入念な RFI テストを義務づけさせたり、ポケットが深くない限り、複雑すぎて 1 から実装するのが実質的に不可能な「合意に基づく」規格を作ったりするわけだ。参入障壁がいちばん強力な市場は、革命家、たとえばインターネットや、司法省対ベルシステムの独禁法裁判などからいちばん攻撃を受けやすい市場でもある。）

参入障壁の低いオープンなプロセスは、分離させるより、むしろ参加をうながす。分離による高いオーバーヘッドなしに成果が得られるからだ。その成果は、分離したら達成できるものほどはすごくないかもしれないけれど、必要コストは低いし、多くの人はいかにがまんできる範囲のトレードオフだと判断する。(スペイン政府がフランコの反バスク法を廃止して、バスク地方に独自の学校と一定の自治権を与えたら、バスク独立運動のほとんどは一瞬で消えた。それでは不十分だと固執したのは、ゴリゴリのマルクス主義者だけだった。)

[RP] 非公式パッチについてはいくつか細かい点がある。まずこれは、「友好的なもの」と「敵対するもの」に分類できる。「友好的」なパッチは管理者のコントロール下で、プロジェクトのソースコードの主流に織り込まれるように設計されている(それが実際に織り込まれるかはまた別の話)。「敵対」パッチは、管理者の認めない方向にプロジェクトをねじまげようとするものだ。一部のプロジェクト(とくに Linux カーネルそのもの)は友好パッチにはすごく甘くて、むしろベータテストの段階では、独自のパッケージを推奨したりさえする。一方の敵対パッチはオリジナルと張り合おうという決断を示すもので、深刻な事態となる。敵対パッチを大量に管理するようになると、プロジェクトはやがて分裂に向かうことが多い。

[LW] ハッカーと海賊ソフト文化との対比がいかに示唆的かを指摘してくれた点で、Michael Funk (mwfunk@uncc.campus.mci.net) には感謝したい。Linus Wallej は、かれらの文化力学についてよくとは異なる分析をポストしている(かれらを希少性文化として描いている。これは *A Comment on 'Warez D00dz' Culture*⁶ を参照。

この対比は、長続きしないかもしれない。

もとクラッカーの Andrej Brandt (andy@pilgrim.cs.net.pl) の見解によると、cracker/warez d00dz 文化は現在しばみつつあって、いちばん優れた人材やリーダーたちは、オープンソースの世界に取り込まれているという。この視点について独立した証拠としては、「Cult of the Dead Cow」というクラッカー集団が 1999 年 6 月に行った、前例のない行動がある。かれらは、Microsoft Windows のセキュリティを破るためのツールである「Back Orifice 2000」を GPL のもとで公開した。

[HT] 進化論的にいえば、職人の衝動自体が(内面化された倫理と同じく)裏切りの高リスクと

⁶http://www.df.lth.se/triad/papers/Raymond_D00dz.html

コストの結果かもしれない。進化心理学者たちは、人間が社会的なだましを検出するのに特化した脳論理回路を持っているという証拠を得ている [BCT]。そして、われわれの先祖がインチキを検出する能力を持ったために選別されたのはなぜか、理解するのは容易だろう。だから、もしリスクもコストも高いけれど、メリットももたらすような性格の持ち主だという評判を得たければ、そういう性質があるふりをしてごまかすよりも、実際にそういう性質を身につけるほうが戦術的に優れているかも知れない（「正直がいちばん」というやつだ）。

進化心理学者たちは、これで酒場のけんかのような行動も説明できるのでは、と言う。若い大人の男性の間では、「タフだ」という評判は社会的にも（そして現在のフェミニズムの栄養が強い風土にあっても）性的にも有用なのだ。「タフさ」を詐称するのは、非常にリスクが高い。それがばれたときのマイナスの影響は、それを主張しなかった場合に比べてもっと悪い立場にその人物を追いやってしまう。この詐欺のコストはあまりに高いので、「タフさ」を内面化して、けんかで無理にそれを証明するために深刻なけがのリスクを負うほうが、最小効用の最大化（ミニマックス化）のためには有効なことさえある。もっと議論にならない「正直さ」のような性向についても、似たような観察がなされている。

創造的な仕事の、主に瞑想にも似た見返りは過小評価してはならないけれど、職人の衝動というのは、おそらく部分的にはまさにこうした内面化からくるものなのだろう（ここでは基本的な性向は、「面倒な仕事をこなす能力」といったようなものになる）。

[MH] マスローのヒエラルキーと関連理論についての簡潔なまとめは、web上で読める。*Maslow's Hierarchy of Needs*⁷

[DC] しかしながら、リーダーに謙虚さを求めるというのは、贈与文化、過剰文化のもっと一般的な性質なのかもしれない。David Christie (dc@netscape.com) は、フィジーの周辺部の島を旅したときについてこう書いている：

「フィジーの村長にも、あなたがオープンソースのリーダーについて述べたのと同じように、卑下する腰の低いリーダーシップのスタイルが観察されました。[中略] 大いなる尊敬を集め、そしてフィジーにある限りの現実の力をすべて持っているのに、われわれが会った村長たちは本物の謙虚さを示し、しばしば自分たちの任務を聖人のように受け入れていました。これは、村長というのが世襲だということを考えてなおさら興味深いものです。選出されるわけでもないし、人気投票で決まるのでもありません。どういうわけか、生まれついでのこと仲間選ばれたわけでもないのに、文化そのものによって訓練されているらしいのです。」

かれはさらに、このフィジーの村長に特有のスタイルは、強力を強制するのがむずかしいという事実からくるのだと思っていることを強調している。村長は「大きなニンジンも、大きな棒も持っていない」わけだ。

⁷<http://www.valdosta.peachnet.edu/~whuitt/psy702/regsys/maslow.html>

[NO] 見ればすぐにわかることだが、うまくいったプロジェクトを創始した人のほうが、成功したプロジェクトのデバッグや援助で同じくらいの仕事をした人にくらべて、高い評判を獲得する。この論文のかつてのバージョンでは、次のような疑問を提示していた：「これは比較できる作業の価値評価として合理的なものだろうか、それともぼくたちがここで示してきた無意識の領土モデルからくる二次的な副産物なんだろうか？」 何人かのコメントは、説得力ある、基本的には同じ理論を示唆してくれている。以下の Ryan Waldron (rew@erebor.com) の分析がこの論点をうまく表現している：

「ロック式土地理論の文脈では、新しい成功したプロジェクトを確立した人物は、本質的に新しい領土を開拓して、そこをほかの人たちが入植できるようにしたことになる。ほとんどの成功したプロジェクトでは、収益はだんだん減るというパターンがあるので、しばらくすると、プロジェクトへの貢献のクレジットは実に関が空いたものになってきて、後からの参加者には、十分な評価が蓄積しにくくなってしまふ。その人がどんなに高質な仕事をしようとも。

たとえば、ぼくがこれから Perl を改善して、Larry や Tom や Randall などが達成した認知のごく一部でも獲得しようと思ったら、本当にすさまじくいい仕事をしなくてはならないだろう。

でも、新しいプロジェクトが [だれか別の人によって] 明日にでも創始されたら、そしてぼくが初期の熱心な参加者になったら、そのうまくいったプロジェクトによって生じた尊敬の分け前をもらう能力は、早めに参加していたということによって大きく拡大する（貢献の中身が同程度だったとしても）。たぶん、早めに Micro\$oft 株を買った人と、後で投資した人の差に近いと思う。みんな儲かるけれど、早めに買った人はもっと儲かる。だからどこかの時点で、ぼくは既存の企業の株が絶えず値上がりするのに加わるよりは、新しい成功する IPO のほうに興味をおぼえるようになるはずだ。」

Ryan Waldron のアナロジーはもっと拡張できる。プロジェクトの創始者は、他の人に受け入れられるか、つかいものになるかもわからない新しいアイデアを、伝道師のように売り込まなくてはならない。だから創始者は IPO リスクと似たようなものを負担することになる（自分の評判へのダメージの可能性として）。これは、仲間内である程度受け入れられたプロジェクトを手伝った人よりも、負担するリスクが大きい。創始者の見返りは、実際問題として助手たちが仕事をたくさんつぎ込んでいても、一定だ。これは交換経済におけるリスクと見返りの関係として楽に理解できる。

他の評者たちの見解では、われわれの神経システムは、ちがいを認識するように調整されていて、安定状態には敏感ではない。新しいプロジェクトの創造によって目的される革命的な変化は、したがってたえまない段階的な改善の集積効果より、ずっと認知されやすい。だが

ら Linus は Linux の父として尊敬される。その他何千という貢献者たちによる貢献の純粋量は、Linus 一人の仕事なんかでは絶対に不可能なほど、OS の成功に貢献してはいても。

[HD] 「脱・共有化」(de-commoditizing) ということばは、「ハロウィーン文書」⁸への言及。この文書でマイクロソフトは、顧客を収奪できるような独占的囲い込みを維持するための、いちばん有効な長期戦略として、この「脱共有化」を何のためらいもなく提案している。

22 謝辞

Robert Lanphier (robla@real.com) はエゴのない行動についての議論に大きく貢献してくれた。Eric Kidd (eric.kidd@pobox.com) は、個人崇拜を避ける意味で謙虚さが大事であるという点を明確にしてくれた。全世界的な影響についての章は、Daniel Burn (daniel@tsathoggua.lab.usyd.edu.au) からのコメントに触発された。Mike Whitaker (mrw@entropic.co.uk) は文化への順応に関する議論の大筋の発想を与えてくれた。Chris Phoenix (cphoenix@best.com) は、ハッカーたちは他のハッカーを追い落としても評判を高めることはできないという事実の重要性を指摘してくれた。

23 改訂履歴

この論文に書かれた内容については、あらゆる誤りや誤解をふくめてぼくが全責任を負うものである。でも、ぼくはコメントやフィードバックを歓迎してきたし、それらを利用してこの論文をよりよいものにしてきた。このプロセスは、どこか決まった時点で止めたりするつもりはない。

1998 年 4 月 10 日 Web 上でバージョン 1.2 を公開。

1998 年 4 月 12 日 バージョン 1.3。誤植の訂正と、最初の一般コメント群への対応。書誌の最初の 4 点。匿名で寄せられた、評判インセンティブが職人自身も気づかないうちに機能しているという見解を加筆。warez d00dz との示唆的な対照性について加筆、「ソフトは自ら語るべきだ」との考えについての部分加筆、個人崇拜の回避に関する考えを加筆。こうした変更の結果、「エゴの問題」の章が拡大して分裂。

1998 年 4 月 16 日 バージョン 1.7。「グローバルな意義」についての章を追加。ヌースフィアの植民地化に関する歴史的トレンドを論じ、「カテゴリーキラー」現象について検討。今後の研究課題を追加。

1998 年 4 月 27 日 バージョン 1.8。書誌に Goldhaber を追加。Linux Expo 予稿集に載るのはこのバージョン。

1998 年 5 月 26 日 バージョン 1.9。Fare Rideau のノウアスフィア・エルゴスフィアの区別を加筆。RMS の、自分は反商業主義ではないという主張を追加。「文化適応とアカデミズム」に

⁸<http://cruel.org/freeware/halloween.html>

ついでに章を追加 (Ross J. Reedstrom、Eran Tromer、Allen MacInnes 他に感謝)。謙虚さ (「エゴのない行動」) について加筆、Jerry Fass と Marsh Ray からのコメントに基づく。

1998年7月11日 バージョン 1.10。本人の示唆に基づき、Fare Rideau のコメントから名声ということばを除く。

1998年11月21日 バージョン 1.14。ちょっとした編集上の修正と古いリンクの訂正。

1999年8月8日 O'Reilly の本用に大幅改訂。Michael Chastain からもらった、プロジェクト分裂のコストと非公式パッチのコストについてのアイデアを組み込む。Thomas Gagne (tgagne@ix.netcom.com) は、「年功が勝つ」というのとデータベースのヒューリスティクスの類似性に気がついた。Henry Spencer の政治とのアナロジー。Ryan Waldron と El Howard (elhoward@hotmail.com) は、目新しさの価値について考えを出してくれた。Thomas Bryan (tbryan@arlut.utexas.edu) は「抱き込んで拡張」に対するハッカーの嫌悪を説明してくれた。Darcy Horrocks は新しい章の「どれほどすばらしい贈り物？」の示唆を与えてくれた。マスロー式価値のヒエラルキーとの関連についての新しい文章と、能力に対する攻撃に関するタブーについて加筆。⁹

⁹ 翻訳上のミスについて、白田秀彰氏 (hideaki@leo.misc.hit-u.ac.jp) からご指摘をいただいた。伏して感謝する
訳者記す