

オープンソースソフトウェアの ビジネスモデルの研究

Study on Business Models of Open Source Software

高橋 道郎

Michio Takahashi

前田 和昭

Kazuaki Maeda

1. はじめに

1999年米国ラスベガスでのコムデックス (COMDEX: Computer Dealer's Exhibition)¹⁾でリーナス・トーバルズ (Linus Torvalds) はマイクロソフト創業者であるビル・ゲイツと並んで基調講演をつとめた。トーバルズが中心となって開発したオープンソースソフトウェア (Open Source Software: 以下OSSと記す) であるリナックス (Linux) が、世界の注目を浴びているために招待を受けたわけである。トーバルズはUNIXに似たオペレーティングシステム (Operating System) の開発に独力で取り組んでいたが、インターネットを介し多数の同好の士と協力し、最終的には大変完成度の高いものを開発した。このオペレーティングシステムにIBMが着目し同社のサーバ機に採用とのうわさが広まり、コムデックスへの登壇となったわけである。このLinuxはOSSの代表とされ、インターネットを介してソースコードが公開され、それを見た全世界の開発者が自発的に改良作業に加わり、その改良されたソースコードをまた公開するという仕組みで完成度が高められた。これに類した試みはそれ以前にも行われていたが、IBMが興味を示すほどの完成度に達したことがニュースとされた。無償のボランティアが開発したものが商用機に必要な信頼性を有するのだろうか、開発費を誰がどのように負担するのだろうか、といった疑問は当初から投げかけられていたが、本格的普及が始まって10年が経ったのを機会にOSSのビジネスモデルを改めて検証し、OSSの発展に資するのが本論文の目的である。

2. OSSの勃興と普及

初期のコンピュータにおいてはハードウェア毎に専用のソフトウェアが求められ、ソフトウェア単体の価格は存在していなかった。オープンソースという言葉は、フリーソフトとの対比で語られることが多いため、両者の違いから説明されがちであるが、ここではビジネスモデルとの関わりに注目し、高橋茂氏の文献ⁱ⁾を参考にIBMのビジネスモデルから話を始めたい。

IBMは1911年、CTR (Computing-Tabulating-Recording Company) として創業、1924年IBM (International Business Machines) と社名を変更した。創業以来パンチカード機械を賃貸はするが、販売はせず、そのレンタル料金に一括して保守と使い方の指導を含めるビジネスモデルをとってきたⁱⁱ⁾。1952年米国司法省はこのビジネスモデルを独占禁止法違反として告訴、1956年和解が成立して、同意審決 (Consent Decree)ⁱⁱⁱ⁾が発効した。同意審決はコンピュータにも適用されたが、当時はまだソフトウェアの価格分離までは要求されていなかった。しかし、この審決は後年のアンバンドリング (unbundling) の下地となった。1960年代の初めまではIBMの大型から小型まで6系統の計算機はそれぞれ別のオペレーティングシステムのもとで動いていたが開発や保守のコストが膨大となるため、1964年に機種を選ばずに使用できるシステム360シリーズを発売した。これによって顧客はハードウェアを買い替えた時にもこれまで使っていたソフトウェアを使い続けることができるようになった。また、X-Yプロッターなどの周辺機器を他社が開発しIBM機に接続運用できるようにするため、IBMは接続仕様を公開した。このため多くの企業がIBM機に接続できる周辺機器のビジネスに参入したが、この動きはコンピュータ本体にも拡大しPCM (Plug Compatible Machine)、互換機ビジネスの隆盛を招いた。IBMがコストをかけ開発したソフトウェアがソフトウェア開発費の負担がない安価な互換機上で動くことは、IBMにとっては痛手であり、IBMはそれまで無償で提供していたソフトウェアを有償にする方針変更に踏み切った (1969年決定、70年1月から実施)。この時点ではオペレーティングシステムはハードウェアの一部とみなされ価格分離されず、パブリック・ドメイン (Public Domain) にあるとされ、PCMメーカーはIBMが開発したオペレーティングシステムを自由に使うことができた。しかし1979年発売の新しいオペレーティングシステムからIBMは価格分離策をとり、それ以降の新しいオペレーティングシステムは有償となった。また、有償となる以前から、顧客に渡されるソフトウェアはオブジェクトコードのコピーであり、ソースコードが渡されることは、特別な契約を結ぶ場合以外にはないと言われてきた。このことはプログラムを作る側と、使う側がはっきりと分けられる様になったともいうことができる。

しかし、商用ソフトウェアが普及する以前は、大学や研究機関で開発したプログラムをソースコードと一緒に公開することが当然のように行われてきた。プログラムを

作る側と、使う側がはっきりと分けられているのではなく、ある時は作る側の人間として働き、ある時は使う側の人として働いていた。公開することによって知識が共有されるだけでなく、多くの人の目にソースコードが触れることにより、改良が加速されプログラムの完成度が向上する効果があった。商用ソフトウェアが全盛時代にあってもこの活動はStallman等により継続されてきたが、代表的な成功例はヘルシンキ大学のLinus Torvaldsが手掛けたLinuxである。彼は自分で開発したUNIXの改良版ともいべきオペレーティングシステムをインターネット上でソースコードを含めて公開した。インターネットを介して世界中のプログラマが一つの研究室に所属しているともいべき環境が造り出され、インターネット上のプログラム開発者は、公開されたソースコードを見ながら、足りない機能を次々に追加して、また公開した。このようにして公開されたプログラムは次々と改良され、いつの間にかビジネスで使う事の出来る信頼性、品質を備えるようになっていった。一学生が趣味で始めたLinuxは、今では多くの企業においてビジネスの中心的役割を演じるようになってきている。

オープンソースという言葉はフリーソフト運動が困難に直面した時に、それを乗り越えるためにEric Raymondらによって使われ始めた用語^{iv)}で、Torvalds が使い始めた語ではないが、オープンソースの成功例の筆頭はLinuxである。

3. OSSのビジネスモデル

3.1 OSSのビジネスモデルに関するこれまでの研究

無償で提供されるOSSはビジネスとして、果たして成り立つのだろうか？そもそもビジネスを構成する一要素としての、品質を期待しうるものであろうか？このような疑問はOSSが世に出て以来、常に投げかけられてきた。この疑問に対する満足な解答は未だ存在しないが、いくつかの研究がなされている。

(1) 京都大学の末松は自著^{v)}において、経済学的観点からオープン化がビジネスモデルを変えると述べている。すなわち、プロプライエタリー²⁾の最初にして最大の成功者であるMicrosoft社ですら、オープン戦略への転換をすでに行っている。ITハイテク企業も次々とOSSを採用し、自らはソリューションを提供する企業に変身している、との説である。この説それ自体は現状並びにその後の進展を正しく認識しているといえるが、OSSについては誰かが開発して利用者に与えられたものであるとの立場である、すなわち所与の案件としての扱いにとどまっており、OSSそのものに対する踏み込みには至っていない。

(2) 竹田は、OSSをビジネスの観点から分類すると次の3種、すなわち「寄生モデル」「共生モデル」「一体化モデル」に整理できる^{vi)}としている。「寄生モデル」とは

OSS自体を販売や開発するのではなく、OSSを利用したいと言った顧客にOSSを導入する支援を行ったり、要員の教育を代行したり開発したシステムの保守を行うことを業務とするビジネスである。このビジネスモデルは参入が比較的容易である反面、大きな事業には発展することが難しい制約をもっている。またOSSの特徴である「多くのボランティア開発要員が参画」に直接寄与する訳ではない。しかしながら対象とするOSSを利用する人を増やし普及させる効果は大きい。

「共生モデル」とは自社のビジネスにOSSを本質的に取り込み、差別化を図り戦略的優位性を獲得しようとのモデルである。Netscapeが自社のブラウザをOSSとして公開し、Microsoft社の製品に対抗しようとした動きや、IBMをはじめとする電機各社がLinuxをサーバ機のオペレーティングシステムに採用したことを指している。このモデルが当てはまるOSSの数は少ないが、OSSのビジネスモデルの主流を占めているので後程改めて述べる。

「一体化モデル」は、小規模なアプリケーションに限られるようだが、ある業界に特化したアプリケーションをOSSとして公開し、業界の標準的な業務システム、プラットフォームとしての地位を狙うモデルとしている。

この3分類による整理は、OSSを「ソフトウェア開発者」と「ユーザ事業者」との2面から考えると、後者の観点からは妥当な分類であるが、前者であるソフト開発者の参画動機を解明するものとは言えない。

(3) 電子商取引のシステムインテグレーターであるOptaros社³⁾はオープンソース・ビジネスの主力となっているビジネスモデルを4種に区分している^{vii)}。

最初のモデルはProprietary Offeringsと称する。ソフトウェアのオープンソース版は一連のプロダクト・ラインのエントリー用として無償で提供され、それとは別に、一般企業が必要とする追加機能、ないし機能強化版が有償のライセンスで提供される。

2番目のモデルはDual Licensesという。オープンソース版もコマーシャル版も同一のソースコードからなり、オープンソース版は無償のGPLなどでその普及を目指し、有償のコマーシャル版はサポートなどの付加価値を付けたものである。

3番目のモデルSubscriptionsは、オープンソースのサービスにかかる購読料方式で、ソフトウェア自身の価格はない。利用するユーザは、パッケージングや新バージョンのリリース、修正パッチ、サポートなどのために年間利用料を支払う。先に述べた竹田の「寄生モデル」に相当する。

4番目のモデルValue Add Servicesは、OSSに対する付加価値サービスである。このサービスは、どれか1つのOSSというよりも顧客の抱えるプロジェクトに対するコンサルティングやインテグレーション、サポート、ホスティングなどを通した総合的なソリューション提供である。Optaros社やLinagora社がこの範疇である、

とっているが竹田の「共生モデル」と近い。

Optaros 社は多くの OSSに通じていることをこの 4 分類を記載したカタログで誇示しているが、先の竹田の研究と同じく、OSSに開発ボランティアが集う仕掛けや仕組みに関する言及はない。

OSSに開発ボランティアが集う仕掛けや仕組みに関する研究は、これまで活発ではなかったといえる。

(4) クリス・アンダーセン氏によれば、フリーの形態には次に掲げる 4 種類がある^{viii)}。

- ・フリー 1：直接的内部相互補助—消費者の気を引いて、他のものも買ってみようと思わせる商品
- ・フリー 2：三者間市場—コンテンツやサービスなど二者間が無料で交換することで市場を形成し、第三者があとからそこに参加するためにその費用を負担
- ・フリー 3：フリーミアム—有料のプレミアム版に対する基本版を無料で配布し、有料版の収益で無料版を支える
- ・フリー 4：非貨幣市場—注目や評判、喜びや満足感など、貨幣以外のものを対価として得る

(5) Karl Fogel は著書「オープンソースソフトウェアの育て方」^{ix)}でこのように述べている。オープンソースプロジェクトが外部からの支援を受ける理由はいろいろあるが、つぎのような理由のどれか、またはいくつかが支援をする理由である。

- ・コスト負担を共有する
- ・サービスを拡大する
- ・ハードウェアの営業をサポートする
- ・競争相手を弱体化させる
- ・宣伝効果をねらう
- ・デュアルライセンスで売り上げをたてる

この 6 項目についての説明は省略するが以下の節においては、3.1 節で掲げた論点を参考にしつつ、いくつかの OSS について、そのビジネスモデルを検討する。

3.2 Linux のビジネスモデル

Linux の誕生と発展については 2 章で述べたので、ここでは Linux 自体が開発を継続している仕組み、すなわちビジネスモデルを記す。Linux は多くのボランティアが開発を自律的に分担していることはよく知られている。Linux Foundation によると、

システムの中核を構成する Linux カーネルは、ソフトウェア開発の歴史の中でも特筆すべき大規模な共同作業の成果です。カーネル修正の数はもともと大きな数に上っていましたが、その数はさらに増え続けています。最近のカーネル公開版には10,000におよぶパッチが含まれています。それぞれのカーネル版の開発には、200近くの企業の1,000人以上の技術者が参加しています。

と大規模な共同作業が行われていると謳っている。このような大規模なプロジェクトを運営する仕組みは、もっぱらLinux Foundationが担っているが、その使命は、Linuxの普及、保護、標準化を進めるために、オープンソースコミュニティに資源とサービスを提供することであり、そのために

- (1) 中立的な協業と教育の場の提供
- (2) Linux 開発の保護と支援
- (3) 技術プラットフォームとしての Linux の強化 (Linux Standard Base (LSB) と Linux Developer Network)
- (4) さまざまの関係コミュニティニーズへの対応。具体的には
 - (ア) 開発者の旅費支援
 - (イ) 守秘義務契約
 - (ウ) Linux 商標管理
 - (エ) リージョナルプログラム

などを提供している。これらの費用はLinux Foundationに加盟している企業の会費で賄われているとされている。また、組織としては16名の理事、同じく16名のスタッフで運営されている^{x)}。理事のうち2名は日本企業である富士通と日立製作所から選任されている。法人会員がLinuxの維持発展に貢献していることを象徴している。このことは、Kroah-Hartmanによると、カーネルの改良への貢献は、個人：13.9%、不明：12.9%、企業：73%、とされていることからもうかがえる^{xi)}。ちなみに改良への貢献が大きい企業は上位から、Red Hat, Novel, IBM, Intelなどである。

IBMは1998年にLinuxに対する調査研究を始め、翌年にはIBMとしての方針を定めた。それは、IBMオープンシステム開発担当VPのDan Frye氏によると^{xii)}、

- (1) IBMのすべてのサーバでLinuxが稼働すること
- (2) IBMのソフトウェアがLinuxをサポートすること
- (3) Linuxに関わるサービスを提供すること

である。現在は600名以上のLinuxとオープンソースの開発者が社内には存在することである。このようにLinuxに肩入れしている背景には、単にLinuxが優れたオペレーティングシステムであり今後の進化が期待できるからだけではない。Microsoft社がPCのオペレーティングシステムとしてWindowsをもち、それが市場をほぼ独占している状況から何とか逃げ出したい、との切実なIBMの事情がそこにはある。サーバにWindowsは不要との意見もあるが、PCの性能向上が著しく従来のUNIXサーバがPCサーバに代替されるようになってきていることも背景にある。

また、半導体メーカであるルネサス・エレクトロニクスもLinuxをサポートしている。(v2.6.30以降のカーネル開発貢献ランキングで9位に入っている)。ルネサスの主力製品であるSHマイコンは組み込みシステムとして自動車や通信機器に多く使われているが、顧客企業における開発環境がLinuxベースに移行していることを反映し、積極的にLinux対応を進めている^{xiii)}。

3.3 Rubyのビジネスモデル

Rubyはまつもと氏の開発したプログラム言語である。同氏を取材したBruce氏の著書^{xiv)}によると、

まつもとゆきひろ氏は、1993年にRubyを開発した。Rubyは、いわゆるスクリプト言語のグループに属し、インタープリタ形式、オブジェクト指向、動的型付けなどの特徴を持つ言語だ。

Rubyは1990年代に趣味で作られ、細々と使われていた。最初の版から10年ぐらいたったところで、大変革が起きた。Rubyは、実行速度という点ではそれほど効率的ではないが、プログラマの生産性を大幅に高めているインタープリタ形式で動作することを前提に開発された。バージョン1.8までがこれにあたる。しかし、実行速度向上の要求があり、Javaと同じように仮想マシンのコードにコンパイルし実行するようになった。Javaで記述した場合と比べて、Rubyだと短い記述ですむ。また、実行時にプログラム自身の情報を簡単に取り出す事ができ、プログラム上の決定(変数に束縛されるオブジェクトのタイプなど)を実行時まで遅らせる事ができるので、上級開発者に受けが良いようだ。Bruce氏によると、まつもと氏は

1993年にPerlを初めて見たとき、なぜか「Lisp, Smalltalk, Perlの特性を組み合わせたオブジェクト指向言語があれば、生産性を高めてくれる素晴らしい言語になるのでは」というインスピレーションが湧いたのです。それで、そのような言語を開発し始め、Rubyと名付けました。

と述べている。

1990年代に主流であったPerlは、便利ではあるものの、記述したプログラムが美しくなく、オブジェクト指向プログラミングに非対応だったので、他の「何か」を探していた開発者達にRubyが受け入れられた。2006年頃、Rails (Ruby on Rails) フレームワークの登場で一気にブレイクした。世の中に拡散するには、必ずキラーアプリケーションが必要である。Rubyの場合は、Railsがキラーアプリケーションであった。Bart Eisenbergによると^{xv)}、まつもと氏は

もともとの動機は、単にプログラミング言語を作りたい(中略) Rubyは企業が設計した言語ではありません。あくまでも私が趣味として作成したものですし、そもそもの意図は、自分がしていたことをして自分が幸せになることでした。

と述べている。Rubyはまつもと氏が趣味で開発し、無料で配布している。その心意気に賛同する開発者達が、共同で開発を進め、最新版も含めて開発されたRubyは、全て無料で配布されていて、そこにはビジネスに関する事は何もない。初期のLinuxと同じともいえるが、これがLinuxと同じ道を追うのか、それとも別の道をたどるのかはまだ分からない。

3.4 YouTubeとWikipediaのビジネスモデル

OSSのビジネスモデル研究になぜYouTubeやWikipediaが出てくるのかを訝しく思う向きも多いであろう。確かにこれらはOSSではない。しいて共通点を挙げれば

- (1) 不特定多数の開発者や投稿者によってソフトウェアやビデオライブラリーが作られている。
- (2) そのソフトウェアやビデオライブラリーの改良や蓄積を育む仕組みというかビジネスモデルが存在し、その巧拙が発展を左右している。
- (3) 言わずもがなのことであるが、インターネットを介して多くの人がソフトウェアの開発改良やビデオライブラリーの集積に寄与できる仕組みとなっている。

などのことが言えよう。この研究はOSSのビジネスモデル、特にOSSによって既存の企業のビジネスモデルがどのように変化するかではなく、OSS自体が持続発展をするためには、そのビジネスモデルは如何にあるべきかを追求するものであるから、このようにOSSと類似点の多いものを取り上げる所以である。

よく知られているようにYouTubeはGoogleの提供するサービスの一部である。であるからここではGoogleのビジネスモデルについて触れる。同社は当初、サーチエンジンの改良に注力し、十分な数の利用者を獲得するまで待ったのちに、検索連動型広告のビジネスを始めた。2010年には社員数が24,400人、全世界での売上は293億2,100万ドル規模にまで達している。売り上げの9割は広告料といわれている。YouTubeはGoogleの提供するサービスの一部であって2006年にGoogleに買収された。Googleの経営層（理事9名、経営管理6名）を見る限り、業界でも有数のテクノロジープロフェッショナルで構成され、世界一の検索エンジンを開発しビジネスに打って出た創業時の精神は大企業になった現在でも受け継いでいることが窺われる。

繰り返しになるが、Googleのビジネスモデルは「世界一の検索サービスを提供することによって多くの人の利用を促す。収入は検索連動型の広告がまかなう。」である。

YouTubeと似ているが、収入源が全く異なるモデルは非商業主義を貫くWikipediaである。そのトップページには「ウィキペディアはオープンコンテンツの百科事典です。方針に賛同していただけるなら、誰でも記事を編集したり新しく作成したりできます。」とあるように記事を作成するのは従業員ではなく一般のボランティアである点で、OSSと通じるところがあり、今回の検討対象に加えられた。

Wikipediaは創業者のJimmy Walesによると、WikipediaはGoogleのサーバ100万台、Yahooのスタッフ13,000に対し、はるかに少ない679台のサーバと95人のスタッフで運営しながら毎月4億5千万人の利用者を有する世界第5位のページビューを誇っている。この95名のスタッフは先に述べたように、記事を作成するのではなく全世界のボランティアが記事を書き、それを改良する作業がはかどるように環境を整える役割を担っている。Wikipediaの活動資金は100%寄付であって、その額は年間600万ドルと言われている。この費用でサーバの管理運営とスタッフの諸経費を賄うのは簡単なことではない。これはクリスマスシーズンの恒例となっているが、WikipediaページのトップページにあるJimmy Walesからの「寄付のお願い」を目にした人は多いと思う。

4. 考察とビジネスモデル研究法の提案

4.1 考察

OSSの普及に伴い、情報や通信に関わる企業のビジネスモデルが変化を強いられている。どのような変化が現れ、今後はどのようなになるのか。これは多くの人々の関心を集めたテーマであるが、ここで取り上げたのはそれではなく、OSSそのものを発展させていくビジネスモデルについてである。3.1節で紹介したFogelは「オープンソフトプロジェクトが失敗に終わる可能性は非常に高く、おそらく90～95%くらいと

みていい」と述べている。このセリフは奇しくも【ベンチャー企業の生存率】を彷彿とさせる。OSSとベンチャー企業とは、あるアイデアや技術をシーズとして発展、すなわち事業化を試み活動するところに共通点がある。ベンチャー企業の生存率について日本の統計は少ないが米国の研究によると、創業時に0～19人の小規模でスタートした会社の1年後の存続率は約8割、4年後の存続率は4～5割程度である^{xvi)}。ここではそれぞれの生存率について検討することはさておき、その成長段階に応じたビジネスモデルを考察する。ベンチャー企業の成長段階は一般に

- (1) シード期
- (2) スタートアップ期
- (3) 急成長期
- (4) 安定成長期

と分類される。OSSについては、その成長段階を分類する用語は確立してはいないために、ここで述べたベンチャー企業の成長段階を借用することにしたい。3章で述べた事例をまず、「OSSを開発するプロジェクト」と「OSSを自社のシステムに活用し事業を行う企業」に分ける。実際には後者の企業に属して、前者の一員として活動する者が多数存在するが、ここでは当面前者の研究に的を絞りたい。前者である「OSSを開発するプロジェクト」に該当する例は、3章で掲げたLinux Foundation, Ruby, YouTube, Wikipedia, が相当している。これを更に、「ベンチャー企業の成長期」に当てはめると、「スタートアップ期」に当てはまるのは、Ruby, 「急成長期」または「安定成長期」に当てはまるのが、Linux Foundation, YouTube, Wikipedia, といえるだろう。Linux Foundationの収入源は法人会費、YouTubeの収入源は広告料、Wikipediaのそれは寄付、とそれぞれのビジネスモデルは異なるが、その収入で賄う活動は、利用者や開発者にとって使いやすい情報通信システムの開発、サーバの運用や保守、投稿者や開発者が意欲をもって参画できる仕組みづくりとその維持管理、と類似の業務が多いことが分かる。しかし、「スタートアップ期」のビジネスモデルに関しては、不明の点が多い。

4.2 RugsOn⁴⁾を用いたOSSのビジネスモデル研究法

(1) RugsOnの概要

インターネットの発達とコンピュータの飛躍的な性能向上を背景にして、XML⁵⁾を使って木構造データやグラフ構造データ（以下、総称して 構造データと記す）を表現することが多くなってきた。XMLは、データに意味を付記できるようにしたマークアップ言語であり、W3Cによって仕様が制定されている。XMLで記述されたデータ

であるXML文書は、テキスト形式で表現され、簡単なテキストエディタでデータの追加修正が可能である。

XML文書は、WWW普及の立役者となったHTMLにはない機能を備え、電子商取引などビジネスで広く活用できるように工夫されている。その応用範囲は、アプリケーション間でのデータ交換、データベースでの情報検索、ネットワークを介したデータ交換などに広がった。データの記法が仕様で標準化され、全ての人に公開されていることが広く受け入れられた原因の一つとなっている。

XMLに関連する仕様は、全ての人が自由に無料で使うことができる。XML文書をプログラムで操作するために多数のライブラリが開発され、SAX^{xvii)}やXerces^{xviii)}など高機能で質の高い製品が無料で公開されている。また、本格的なXML文書の設計・編集は複雑な作業となるため、設計からコード生成までの一連の作業を支援する多機能なツールとして、XMLSpy^{xix)}が有料で提供されている。XML文書の規模が拡大化してくると、必要な情報を取り出すための検索機能が必要となり、データの格納・高速検索のためのデータベース管理システムとして、Tamino^{xx)}やNeoCore^{xxi)}などが有料で提供されている。

このように、仕様で標準化されたXMLの「記法」を中心に、「操作」「編集」「保管」が重要な役割を担っている。XMLの「記法」とライブラリによる「操作」は無料で公開されているものの、人手では複雑な作業となる「編集」と「管理」に関する分野でビジネスが展開されている。

XMLには、プラットフォームに依存しない特徴があり、異種のプラットフォーム間でのデータ交換が可能となる。ところが、データ記述の拡充を中心に発展したため、最新のプログラミング言語がXMLデータの処理に向いているとは限らなくなった。また、XML文書には開始タグと終了タグが必要であり、読み書きする人の立場からは、冗長さゆえ面倒なことが多い。また、標準化されている仕様の量が膨大すぎて、全ての内容を把握することは困難である。そこで、最新のプログラミング言語の構文のサブセットを使って、なるべく簡潔な表現で構造化データを記述できるようにし、プログラム開発ツールを提供することにした。その記法をRugsOnと呼ぶ。RugsOnは著者の一人である前田によって研究開発が進められている。

RugsOnは、括弧を使わないメソッド呼び出しと、複数メソッド呼び出しをまとめるためのブロックを使って構造化データを表現する。この記法は、Ruby、Groovy、Scalaの3つのプログラミング言語に共通である。また、グラフ構造データの表現を簡単に記述可能としている。さらには、Rubyを使ってRugsOnによる構造化データの定義を可能とし、プログラミングを簡潔にするためのコード生成ツールの開発も進んでいる。

(2) RugsOn を用いたビジネスモデル研究法の提案

XML との対比で考えると、「記法」のための RugsOn の仕様と、「操作」のためのプログラム開発ツールを無料で広く公開することを予定している。その後、「編集」のための高機能なプログラム開発ツールや「管理」のためのデータベース管理システムを有料で提供することを考えている。これは、第 3.1 節で述べた「フリーミアム」にあたる。大事なことは、有料のプレミアム版に対する基本版を無料のオープンソースとして配布し、広く世界中に行き渡らせ、それを土台にビジネスを展開することである。また、開発者魂を鼓舞するようなソフトウェアを OSS として公開し、外部の開発者に先進的な開発を委ねることで、開発のスピードを向上させることを狙っている。

まず最初は個人で Web サイトを構築するところからはじめ、将来的にビジネス利用に展開できることを考慮しながら Web サイトの充実を進めている。また、独自に設計した RugsOn を世界中のオープンソース開発者に公開するため、世界各地で開催される開発者会議での講演を実現し、オープンソースのビジネス化に向けた広報活動を始めている。開発してビジネス化すべき製品は、プログラム開発ツールとデータベース管理システムである。これまでに構築済みの開発ツールを土台にして、世界中に公開できるレベルにまで品質を向上させる必要がある。

ビジネス化の問題点として、独自のデータベース管理システムを開発するために、時間と費用がかかることがある。オープンソースとしてビジネスを立ち上げるために、公開するソフトウェアのライセンスを自分で保有する必要がある、そのためにデータベース管理システムをゼロから開発することになる。この開発は、短期間だけでは不可能であり、早急に実用化フェーズを設定して、熟練したソフトウェア開発者の雇用、または、優秀なソフトウェア開発会社への外注が必要となるであろう。

5. まとめ

OSS に関わるビジネスモデルの研究を行った。まず OSS によって既存企業の事業がどのように変わっていくのかについての、これまでの研究を紹介した。次いで OSS に関連する Linux Foundation, Ruby, YouTube, Wikipedia, についてそれぞれのビジネスモデルを述べた。Linux Foundation, YouTube, Wikipedia, の 3 者については、その収入に関してはそれぞれ、法人会費、広告料、寄付、と異なっており、得た収入をどのように使うかのコスト構造は規模の差はあるものの、

- (1) 利用者や開発者にとって使いやすい情報通信システムの開発,
- (2) サーバの運用や保守,
- (3) 投稿者や開発者が意欲をもって参画できる仕組みづくりとその維持管理,

と類似の業務が多いことが分かった。しかし、スタートアップ段階にあるプロジェクトのビジネスモデルは不明の点が多い。

この研究を進めるため、前田の開発したRugsOnとそれに関連したOSSを使ってビジネスモデル確立までの追跡調査を行う計画を企画し提案した。

注

- 1) 1979年から2003年まで毎年11月、ラスベガスで開催されていたコンピュータ関連の展示会である。CeBITに次ぐ世界第二位の規模のコンピュータ見本市であり、他の業界を含めても最大級の見本市であった
- 2) プロプライエタリ・ソフトウェア（英: proprietary software）とは、ソフトウェアの使用、改変、複製を法的・技術的な手法を用いて制限しているソフトウェア
- 3) 本社は米国ボストン、従業員150名、アパレル業界に顧客多数。
- 4) RugOn 詳しくは後述するが、簡潔な表現で構造化データを記述できるようにした記法
- 5) [XML] W3C, Extensible Markup Language (XML) , <http://www.w3.org/XML/>

参考文献

- i) 高橋茂, プラグコンパティブルメインフレームの盛衰 (1), 情報処理, 44巻3号, pp.270 - 276 (2003年3月)
- ii) Pugh, E.W. Origin of Software Bundling, IEEE Annals of the History of Computing, Vol.24, No.1, pp.57-58 (Jan.-Mar.2002)
- iii) United States District Court: IBM 1956 Consent Decree, Civil Action No.72-344 (Jan.25, 1956) <http://www.cptech.org/ap/ibm/ibm1956cd.html>
- iv) Eric Steven Raymond, 伽藍とバザール, 山形浩生訳, 光芒社, 1999年
- v) 末松千尋, オープンソースと次世代IT戦略, 日本経済新聞社, ISBN4-562-31146-2
- vi) 竹田昌弘, オープンソース・ソフトウェアとビジネスとの関係に関する考察, 立命館経営学 Vol.44, No.3, pp.53
- vii) 「Open Source Catalogue 2007」, p.7, Copyright©2007 Optaros, Inc. 森洋一氏の訳 (<http://i.impressrd.jp/e/2008/08/07/32>) を参照 (2011.11.22)
- viii) クリス・アンダーセン, フリー, NHK出版, 2009年
- ix) Karl Fogel (高木正弘ほか訳), オープンソース・ソフトウェアの育て方, オライリージャ

バン, ISBN978-4-87311-412-5

- x) Linux foundation HP, <http://www.linuxfoundation.jp/about/staff>, 2011.11.30
- xi) Greg Kroah-Hartman, et.al, Linux Kernel Development, March 2008
- xii) Dan Frye, Linux at IBM (<http://www-06.ibm.com/jp/linux/> 2011.11.21 LinuxCon Japan/Tokyo 2010
- xiii) ルネサス・エレクトロニクス HP, http://japan.renesas.com/products/tools/introductory_tools/sh_linux/sh_linux_tools_product_landing.jsp 2011.11.21
- xiv) Bruce A. Tate 著 まつもとゆきひろ 監訳 田和勝 訳 7つの言語7つの世界 ISBN 978-4-274-06857-7
- xv) Bart Eisenberg, プログラマを幸せにすることを目標にした言語 Ruby, Software Design, 2012年2月号, pp.158-161.
- xvi) 長谷川博和, ベンチャーマネジメント, 日本経済出版社, p91, ISBN978-4-532-13391-7
- xvii) SAX, <http://www.saxproject.org/>
- xviii) Xerces, <http://xerces.apache.org/xerces2-j/>
- xix) XMLSpy, <http://www.altova.com/jp/xmlspy.html>
- xx) Tamino. <http://www.softwareag.com/corporate/products/wm/tamino/>
- xxi) NeoCore, <http://www.neocore.jp/>