特集：
オープンソース
ソフトウェア

オープンソース ソフトウェアの現状

1

青山幹雄
南山大学数理情報学部情報通信学科
mikio.aoyama@nifty.com

オープンソースソフトウェアとは「ソースが公開された」ソフトウェアである．しかし，その意義は，ソフトウェア開発からシステムインテグレーション（SI）に至る広い範囲に及んでいる．オープンソースソフトウェアがこれらの現場で広く利用されるようになっていることから，その本質と効果やリスクを理解しておく必要がある．本稿では，オープンソースソフトウェアの現状を，開発と利用の観点から概説する．

■オープンソースソフトウェアとは

－オープンソースソフトウェアの定義

オープンソースソフトウェアの意味を共通に理解するために，オープンソースイニシアティブ（OSI：Open Source Initiative）[10] ではオープンソースソフトウェアの定義として，単にソースコードが公開されているだけでなく，下記の 9 つの具体的な条件を満たすことを求めている．

(1) 再配布の自由（Free Redistribution）：ソフトウェアを有償，無償を問わず自由に再配布可能であること．

(2) ソースコードの公開（Source Code）：ソースコードを含んで配布されること，あるいは，複製の費用程度を超えない妥当な対価で入手可能なこと．

(3) 変更したソースコード（二次的著作物）の再配布可能（Derived Work）：変更が可能で，かつ，変更したソースコードを同じ条件で再配布可能であること．

(4) 原作者のソースコードの一貫性保証（Integrity of The Author's Source Code）：変更をパッチで配布する場合に限り，ソフトウェアの一貫性を維持するためにソースコードの配布を制限できる．

(5) 個人・団体の差別禁止（No Discrimination Against Persons or Groups）：いかなる個人やグループも差別しない．

(6) 使用分野の差別禁止（No Discrimination Against Fields of Endeavor）：使途の差別をしない．これは，営利目的の利用を含む．

(7) ライセンスの流通（Distribution of License）：ライセンスは再配布する場合にそのまま適用され，追加のライセンスは不要．

(8) ライセンスの特定製品依存の禁止（License Must Not Be Specific to a Product）：ライセンスは特定製品（ディストリビューション）に依存してはならない．ある製品に含まれるソフトウェアを別の製品に含めた場合，ライセンスは製品によらず，そのソフトウェア本来のライセンスが適用されなければならない．

(9) 他ソフトウェアへの干渉禁止（License Must Not Contaminate Other Software）：同梱ソフトウェアなどの他のソフトウェアへ制約を及ぼさない．

これらの条件はオープンソースソフトウェアに関する平均的基準である．後述するように，オープンソースソフトウェアのライセンスはきわめて多様化している．ライセンス条件によって，オープンソースソフトウェアといえどもその扱いに違いがある．特に，近年，ビジネス利用が進んだことから，実用上適用が難しいライセンス条件もある．たとえば，GNU の GPL（General Public License）では派生ソフトウェアも GPL でライセンスすることを厳格に義務付けているが，派生の範囲は明確に規定することが難しい場合がある．そのため，ライセンスによって流通性を損なうおそれがある．これを回避するために，たとえば，Linux ではインタフェースはカーネルに含まれないと宣言し，Linux 上で動作するソフトウェアが GPL に抵触しないようにしている [1]．

－オープンソースソフトウェアの進化

図 -1 に示すように，オープンソースソフトウェアの開発や流通は，大学などの研究者の間では古くから行われてきた．しかし，1980 年代半ばからのフリーソフトウェア活動の台頭と 1990 年代半ばからのオープンソースソフトウェアのビジネス利用の広がり，ならびに，インターネットによる組織を超えた分散協調的開発の実現によって大きな変貌を遂げた [2), 9)]．

「フリーソフトウェア」という言葉は，1983 年に Stallman が出した，"give it away free" と述べたメールに

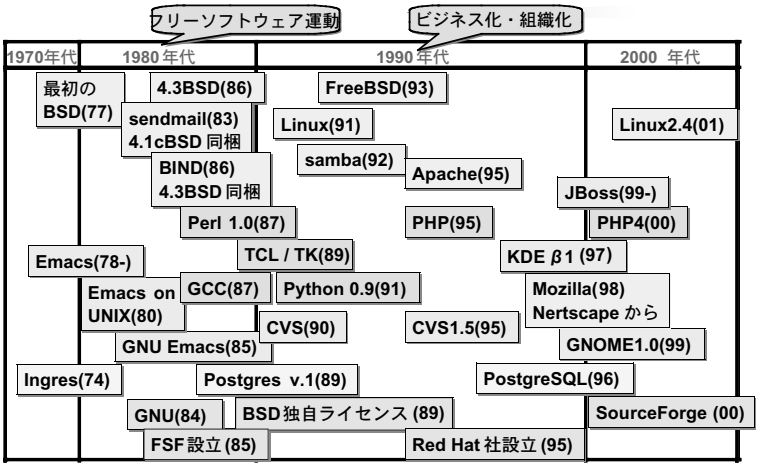

図 -1　オープンソースソフトウェアの進化

名　称	無　償	再配布可	使用制限なし	ソース公開	ソース変更可	コアコード変更公開	成果無償公開
試　供	○（機能制限）						
非営利利用	○（利用形態による）	○					
シェアウェア	○（任意ライセンス）	○					
ロイヤリティフリーバイナリ	○	○	○				
ロイヤリティフリーライブラリ	○	○	○	○			
オープンソース (BSD)	○	○	○	○	○		
オープンソース (Apache)	○	○	○	○	○	○	
オープンソース (Linux)	○	○	○	○	○	○	○

表 -1　オープンソースと関連する概念

端を発している．その後，GNU（GNU is Not Unix）とFSF（Free Software Foundation）の設立とともに，1980年代後半に広まった[4),6)]．この「フリー」は彼自身が述べているように「自由」という意味であった．しかし，英語では「無料」の意味があることから多くの誤解を生んだ．また，「自由」の意味付けが象徴するように，フリーソフトウェア活動は，商用ソフトウェアに対するいわばアンチテーゼとして，文化活動的な側面があった．このような流れに沿って，1980年代半ば以降，大学を中心に多くのフリーソフトウェアが開発され，その中から優れたソフトウェアが流布した．

1990年代半ばには，インターネットの普及によってオープンソースソフトウェアの協調的で組織的な開発が急速に広まった．これは，Raymondの著した「伽藍とバザール」[11)]によって広く認識されるようになり，ソフトウェア開発の新たなモデルとしても注目されている．

一方，ビジネス面では，Linuxの成功とそれによって生まれたRed Hatなどの，いわゆるディストリビューションビジネスの台頭[8)]，さらには，IBMを始めとするメインフレーマなどのオープンソースソフトウェアへの積極的な支援とビジネス活動によって，オープンソースソフトウェアがビジネスとしての地位を確立した．

このような状況の中で，フリーソフトウェアのもたらした意味の混乱を回避するために，それに代わる名称として「オープンソースソフトウェア」が1998年にRaymondらにより提唱され，広く使われるようになった[9),10)]．

―オープンソースソフトウェアに関連する概念

オープンソースソフトウェアの類似の概念として，次のような概念がある．表 -1 にオープンソースのいくつかの代表例とともに示す．

(1) フリーウェア

バイナリコードで無償配布されるソフトウェア．再配布可能であるが変更不可能な場合が一般的である．Stallmanのいうフリーソフトウェアとは異なる．

(2) パブリックドメインソフトウェア

「パブリックドメイン」の概念はソフトウェアに限らず「公共のもの」という意味である．個人の権利を放棄する代わりに責務も負わない．パブリックドメインソフトウェアから商用ソフトウェアを開発することは可能である．

(3) シェアウェア（Shareware）

無償配布され試行できるが，ユーザが利用価値を見出したら，対価を払って利用を継続できる．一般に廉価である．また，通常，ソースコードは公開せず，変更はできない．

国内では，ベクター（www.vector.co.jp）などのフリーウェアとシェアウェアの流通サイトがある．

■主なオープンソースソフトウェア

実際に利用されているオープンソースソフトウェアの例を図 -2 と表 -2 に示す．オープンソースソフトウェア開発の数を推測する1つの目安はオープンソースソフトウェア開発のホスティングサイトであるSourceForge[13)]のプロジェクト数である．ここには50,000を超えるオープンソースソフトウェアの開発プロジェクトがあり，登録ユーザは50万人を超えている．2002年に開設された日本語サイトでも250を超えるプロジェクトが登録されている（2002年11月時点）．もちろん，これ以外にもオープンソースソフトウェアの開発プロジェクトがある．表 -2 に挙げた例は，成功事例の一部である．

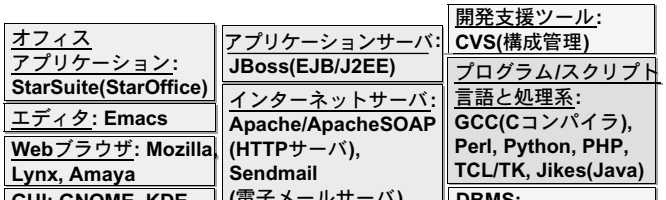

図 -2　オープンソースソフトウェアの例

図 -2 から，オープンソースソフトウェアには次の 2 つの特徴があることが分かる．

(1) オープンソースソフトウェアに適した分野

オープンソースソフトウェアは OS やネットワーク関係などのプラットフォームやプログラミング言語処理系など，仕様が明確なソフトウェアやすでに手本となるソフトウェアが存在している分野に適しているといわれている．現段階では，業務アプリケーションなど，顧客の要求が不明確であったり，顧客個別の対応が必要であったり，要求抽出に多くの工数と技術を要するようなソフトウェアには適さないと考えられている．

(2) インターネット関連におけるオープンソースソフトウェアの地歩の確立

インターネット関連の基盤ソフトウェアに優れたオープンソースソフトウェアがそろっていることから，インターネット関連の基盤ソフトウェアはオープンソースソフトウェアが地歩を確立しつつある．たとえば，Web サーバ用などのアプライアンスサーバは Linux と Apache の組合せが多い．また，Java や XML 関連のソフトウェアではオープンソースソフトウェアが主流であるといってよい．

■オープンソースソフトウェアのビジネスモデル

オープンソースソフトウェアのビジネスモデルをベンダとユーザの観点から整理して示す．

－オープンソースのベンダビジネス

1990 年代後半にオープンソースのビジネスモデルの確立が実際のビジネスを通じて追求された．現段階で，オープンソースソフトウェアのビジネスモデルは製品としてのソフトウェアよりソフトウェアを利用するユーティリティあるいはソリューションビジネスに主眼が置かれている．これは，オープンソースソフトウェアを入手するコストがきわめて廉価であることによる．このようなビジネスモデルは，ソフトウェアの価値をその効用で評価することを意味しており，従来の意味での「モノ」としてのソフトウェアの価値とは異なることから，ソフトウェアビジネスのあり方を問い直すものである．

以下，オープンソースソフトウェアによるビジネスの代表的なパターンを示そう．

(1) ディストリビューションビジネス

さまざまなオープンソースソフトウェアの中からユーザにとって価値があるソフトウェアを選択し，使いやすいようにパッケージ化して提供する．良質のソフトウェアの品揃えがあり，かつ，よく試験され安定しており，インストールなどが容易になっていることが期待される．Red Hat など，Linux をベースとする多くのディストリビュータが 1990 年代後半に出現したが，淘汰を経て，ブランドを確立した少数のディストリビュータが寡占化する傾向にある．しかし，ディストリビューションビジネスの収益性は高くなく，むしろ，以後のサポートビジネスへの入口となると考えられる．

(2) インテグレーションビジネス

オープンソースソフトウェアを組み込んだ SI やソリューションビジネス．適切なオープンソースソフトウェアを組合せて，あるいは，必要に応じて非オープンソースソフトウェアや個別開発ソフトウェアも組み合わせ

分類	ソフトウェア	URL
OS	Linux	www.kernel.org　[日本 Linux 協会：jla.linux.or.jp] [Linux コンソーシアム：www.linuxcons.gr.jp] [Linux Business Intiative：www.lbi.gr.jp]
	FreeBSD	www.freebsd.org [FreeBSD 友の会：www.jp.freebsd.org]
GUI	GNOME	www.gnome.org [日本 GNOME ユーザ会：www.gnome.gr.jp]
	KDE	www.kde.org [日本 KDE ユーザ会：www.kde.gr.jp]
	X Window System	www.x.org
Web ブラウザ	Mozilla	www.mozilla.org
	Lynx	lynx.isc.org
	Amaya	www.w3c.org/Amaya/
エディタ	Emacs	www.gnu.org/software/emacs/emacs.html
アプリ	StarSuite	www.sun.co.jp/software/starsuite/
インターネット	Apache	httpd.apache.org
	Sendmail	sendmail.net
	BIND	www.isc.org/products/BIND/
DBMS	PostgreSQL	www.ca.postgresql.org / index.html [日本 PostgreSQL ユーザ会：www.postgresql.jp]
	MySQL	www.mysql.com [日本 MySQL ユーザ会：www.mysql.gr.jp]
プログラム／スクリプト言語・処理系	GCC(C)	www.gnu.org/software/gcc/gcc.html
	Perl	www.perl.com
	Python	www.python.org
	TCL / TK	www.scriptics.com/software/
	PHP	www.php.net [日本 PHP ユーザ会：www.php.gr.jp]
	Jikes(Java)	oss.software.ibm.com/developerworks/projects/jikes/
開発支援	CVS	www.cvshome.org
ファイルサーバ	samba	www.samba.org 日本 samba ユーザ会 [www.samba.gr.jp]

表 -2　オープンソースソフトウェアとその Web ページの例

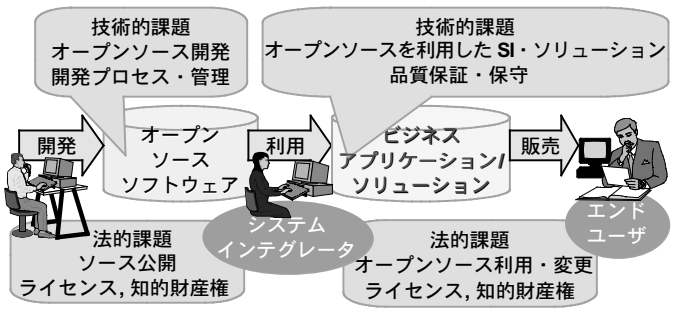

図-3　オープンソースソフトウェアのビジネスと課題

て，ユーザの要求を満たすシステムの提供，ならびに，技術的支援や利用の支援を行う．

(3) 保守・技術サポート

ユーザに対して，オープンソースソフトウェアの保守や問題発生時のコンサルティングなどの技術的サポートを行う．さらに，このような技術者を育成することも支援している．たとえば，Red Hat は RHCE（Red Hat Certified Engineer）という認定制度を設け，技術者の育成を支援している．

(4) 技術の普及促進媒体としてのオープンソースソフトウェアとコミュニティ

従来，技術的には優れていてもユーザや市場に受け入れられなかったために潰れたソフトウェアは少なくない．一方，NFS（Network File System）など，当初はフリーソフトウェアとしてユーザを獲得してから有償化するビジネスモデルが成功してきた．このアプローチを一歩進めて，ベンダが積極的にオープンソースコミュニティに協力することにより，コミュニティは技術の普及やフィードバックの媒体となる．ただし，オープンソースコミュニティ側の受け入れや，そこにおける利益相反（conflict of interests）がないなどの倫理規定をベンダが準拠することが条件となる．

－オープンソースソフトウェアユーザのビジネス効果

オープンソースソフトウェアのユーザはビジネスの観点からどのような利益を期待しているのであろうか？国内でのユーザアンケートがその一端を示している[15]．アンケートを見ると，オープンソースソフトウェアユーザの典型として，次の 2 種類がある．

(1) オープンソースを自前で保守できるパワーユーザ

自らオープンソースを保守する技術力を備え，ソースコードのオープン性のメリットを直接享受する．鉄鋼プラント制御の事例のように長年にわたり自前で保守する場合，ソースのオープン性は重要な要件である．

(2) ブラックボックスとして利用するユーザ

インターネットのアプライアンスサーバのような低価格サーバの場合，ユーザはソフトウェアをブラックボックスとして利用する．ユーザは初期コストの低減に加え，オープン性の間接的な利益として標準への準拠性，信頼性の高さによる TCO（Total Cost of Ownership）の低減と可用性の高さなどのメリットを期待できる．

－オープンソースビジネスの課題

オープンソースソフトウェアビジネスには技術面と法律面で，図-3 に示す，さまざまな課題もある．

■オープンソースソフトウェアの開発

近年，オープンソースソフトウェアの開発方法が多方面から研究されている．このような研究からいえることは，ソフトウェア工学では当然のことであるが，成功しているオープンソースソフトウェア開発では優れた開発プロセスと管理，リーダシップを確立し，実践していることである．しかし，超短期リリースによる高速インクリメンタル開発など，オープンソースソフトウェア開発独特の実践技術も見られる．また，それらの技術は非オープンソースソフトウェア開発でも有効であることが認識されつつある．当然ながら，オープンソースソフトウェア開発がすべて成功しているわけではない．

ここでは，開発プロセス，ソフトウェアアーキテクチャ，開発支援環境，組織とコミュニティの点からその特徴を紹介する[3),12)]．

－開発プロセス

オープンソースソフトウェアの開発は仕様がある程度明確なソフトウェアシステムを対象としてきたことから，開発プロセスは下流工程偏重となる．

一般に，オープンソースソフトウェアの開発プロセスは次のような特徴があるといわれている[3)]．

(1) 高速インクリメンタル開発（超短期リリース更新）
(2) （逐次型開発プロセスより）並行開発プロセス
(3) グローバルに分散した開発者が参画する分散開発プロセス
(4) 独立したピアレビューの実施
(5) 開発者やユーザが構成するコミュニティからの直接的で迅速なフィードバック
(6) 高い技術力とモチベーションを持った技術者の存在
(7) 非オープンソースに比べユーザの積極的で高度な参画

開発プロセスの具体例として，図-4 に示す FreeBSD の開発プロセスの分析例が示されている[5)]．この開発プロセスでも，独立したピアレビューの実施，統合前の適合性試験による統合システムのリスクの回避，開発内部

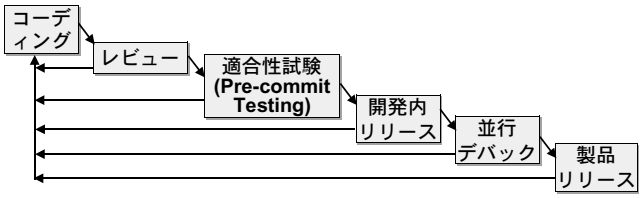

図-4　FreeBSD に見るオープンソースソフトウェアの開発プロセス

リリースと外部への製品リリースの2段階リリース方法など，オープンソースソフトウェア開発の特徴が現われている．オープンソースソフトウェアの開発は分散したボランティアベースであることから，プロセスは管理上のリスクを低減する仕組みを取り入れている．これらの各要素技術は，非オープンソースソフトウェアの開発でも優れた開発プロセスでは実践されているものである．

－ソフトウェアアーキテクチャ

Linux など成功しているオープンソースソフトウェアのモジュールは独立性の高い，良い構造となっている[1),3)]．これは，ソフトウェアアーキテクチャとして優れているだけでなく，分散した組織で並行して開発できるために必須の要請である．また，これを支援する機構として，たとえば，Linux ではプログラムの動的ローディング機構を開発するなど，独立性を高める工夫をしている[1)]．

－開発支援環境

オープンソースソフトウェアの開発支援環境は，グローバルに分散し，かつ，コンピュータ環境などの異なる開発者が共通に利用できる必要があることから，CVS（Concurrent Versions System）などの構成管理ツールを核とし，電子メールや Web といったごく標準的で基礎的な通信手段を用いることが多い．これは，オープンソースソフトウェア開発に参画している開発者の所属する組織の計算環境が異なり，その最大公約数的環境を利用しているためである．このため，核となる開発者が異動すると開発が立ち行かなくなることもあった．

これに対し，オープンソースソフトウェア開発のホスティングサービスとして2000年に SourceForge が開設され，計算資源や開発環境の制約がある程度解消された．SourceForge は，オープンソースソフトウェア開発に留まらず，ソフトウェア開発のホスティングサービスとして興味ある形態である．

－開発組織とコミュニティ

オープンソースソフトウェア開発の大きな特徴は開発組織とそれを取り巻くコミュニティの存在である．

(1) 開発組織

所定の期間でソフトウェアを開発するためには，ボランティア組織であっても組織としての整合性，一体性を保ち，かつ，参画者の技術力や置かれた状況の多様性がもたらすリスクを回避する必要がある．このため，開発への貢献に応じた階層的組織構成をとる．具体例は FreeBSD の事例などを参照願いたい．

開発指示は，企業内開発の，いわゆる上意下達ではなく，実績に基づくリーダシップが基本である．そのため，Linux における Linus Torvalds や Apache の Brian Behlendorf など，核となるリーダの資質が鍵となる[9),14)]．これらのリーダは，通常，ソフトウェアの最初の作者やプロジェクトの創立者である．実際，オープンソースソフトウェア開発の成功の鍵はリーダの資質に大きく依存しているといわれている．

オープンソースソフトウェアの開発組織が大きくなってくると1人では全体の統率が困難となる．そのため，ある段階から，コアメンバと呼ばれる数人程度による集団指導体制へ移行することが多い．たとえば，Apache の場合，1995年に8人の開発者で始まったが，1999年時点でこのプロジェクトに参画した人数はのべ400人を超えている．そのためコアメンバも25人となった[7)]．しかし，開発規模の83%はトップ15人によって行われているように，コアメンバが実質的な開発を担っていることが分かる．

(2) コミュニティ

コミュニティはオープンソースソフトウェアの開発組織の中で特徴的な役割を果たしている．コアメンバ以外の一般の技術者やユーザが誰でも何らかのかたちで参画できるオープン性と，組織としての価値の共有や一体感がコミュニティの特質である．オープンソースの前身としてのフリーソフトウェア運動が，いわば一種の文化活動的な意味合いを持っていた点を併せ考えれば，オープンソースにおけるコミュニティとその文化の果たす役割は重要である．

コミュニティは，開発者コミュニティとユーザコミュニティに分けられるが，両者の間は必ずしも明確に分離できないことが多い．

(a) 開発者コミュニティ

オープンソースソフトウェアの開発には，コアメンバやコミッタなどの定常的に参画している開発者に加えて，アドホックに参画している多くの開発者がいる．その正確な数は不明であるが，Linux の場合，数万人から数十万人と推測されている．

オープンソースソフトウェアの開発者のコミッタの地理的な分布が米国偏重でない特徴がある．たとえば，2000年に実施されたある調査によると，Linux のカーネルの開発者の分布は28カ国にわたり，その中で，

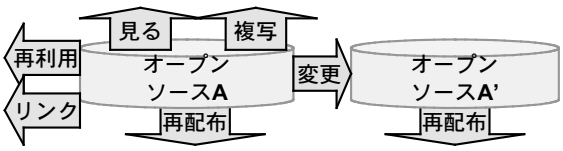

図 -5　オープンソースの利用とライセンス

米国が 45%，ヨーロッパが 40% であった [3]．しかし，我が国からの貢献は必ずしも高くない点は留意すべきであろう．

(b) ユーザコミュニティ

オープンソースソフトウェアの利用がサーバ主体であることを考慮すれば，オープンソースソフトウェアのユーザは企業や大学などの技術者や研究者など，ソフトウェアに関して比較的高度な知識を持った技術者であると考えられる．また，ソースコードのオープン性によりユーザは開発にも貢献できる．したがって，優れたユーザコミュニティを獲得したオープンソースソフトウェアは，ユーザコミュニティからのよいフィードバックが得られ，品質面などで大きな利益を得る．また，新たなソフトウェアの開発においても，一般に，初期のユーザとなり育成者となるのはこれらのパワーユーザである．したがって，新たなソフトウェアを普及させる媒体として，コミュニティの果たす役割は大きい．

■オープンソースソフトウェアのライセンス

オープンソースソフトウェアのビジネスにおいてライセンスの条件は多様であり，また，その法的妥当性が必ずしも明確になっていないことから，大きな課題といえる．

図 -5 にオープンソースソフトウェアを利用・変更する場合のライセンスに関する問題を示す．OSI では，さまざまなオープンソースソフトウェアのライセンスのリストを公開している [10]．

留意すべきことは，OSI のオープンソースの定義で挙げられている再配布や変更などの条件が実際のライセンスではさまざまに組み合わされ，かつ，例外的な条項が付加されている場合もあるため，難解になっていることである．さらに GNU のライセンスでは，LGPL（Library あるいは Lesser GPL）の場合，リンクによるライセンスの伝搬性が謳われ，利用者にとって厳しい制約となり得る点である．実務においても，このようなライセンス条件があることを認識すべきである．また，一方では，ライセンスの法的解釈が確立しているとはいい難く，かつ，実際の係争例がほとんどないことから，ユーザにとっても判断が難しいことも認識する必要がある．

■今後の展望

オープンソースソフトウェアのビジネスへの展開はインターネット関連などで，その地歩を確立しつつある．今後，これらの分野を中心に SI やソリューションへの利用が広がると思われる．また，その内容も OS などの基盤ソフトウェアから，最近では，高度なミドルウェアさえオープンソースで開発されている．

一方，企業の技術普及の戦略として企業とコミュニティの連携やソフトウェア開発のホスティングサービス，ソフトウェア開発のモデルとしてのオープンソースソフトウェア開発プロセスなど，ソフトウェア開発の技術とその普及方法の新たなモデルやあり方として注目すべきものがある．これらの中には，従来の非オープンなソフトウェアの開発に適用できるものもある．また，関連して，政府や自治体などの公共調達におけるソースコードの公開性も課題となるであろう．

今後，オープンソースソフトウェアの開発と利用の両面で，研究・開発，技術者の育成，ビジネスモデルの確立などの推進を図る必要がある．

なお，本稿は，2001 年度に（社）情報サービス産業協会オープンソース調査委員会の活動として行った結果のまとめ [15] に加筆訂正したものである．本調査にご協力いただいた関係各位に感謝する．

参考文献

1）DiBona, C., Ockman, S. and Stone, M.: Opensources, O'Reilly（1999）（倉骨彰訳：オープンソースソフトウェア，オライリー・ジャパン（1999））．
2）Edwards, J.: The Changing Face of Freeware, IEEE Computer, Vol.31, No. 10, pp.11-13（Oct. 1998）．（青山幹雄訳：フリーウェアの台頭と変貌，情報処理，Vol.40, No.1, pp.32-35（Jan. 1999））．
3）Feller, J. and Fitzgerald, B.: Understanding Open Source Software Development, Addison Wesley（2002）．
4）GNU（日本語）：http://www.fsf.org/home.ja.html
5）Jorgensen, N.: Putting it all in the Trunk: Incremental Software Development in the FreeBSD Open Source Project, Information Systems Journal, Vol.11, No.4, pp.321-336（2001）．
6）Lang, C.B.: Freeware, Programming Freedom and Such: http://pauillac.inria.fr/~lang/hotlist/free/index.html.
7）Mockus, A., Fielding, R. T. and Herbsleb, J.: A Case Study of Open Source Software Development: The Apache Server, Proc. ICSE 2000, IEEE CS Press, pp.263-272（May 2000）．
8）日本 Linux 協会（監修）：Linux 白書 2001-2002, インプレス（2001）．
9）Moody, G.: Rebel Code: Linux and the Open Source Revolution, Perseus Publishing（2001）．（小山裕司監訳：ソースコードの反逆，アスキー，（2002））．
10）OSI（Open Source Initiative）：http://www.opensource.org.
11）Raymond, E. S.: The Cathedral and Bazaar（1997）：http://www.tuxedo.org/~esr/writings/（山形浩生訳：伽藍とバザール，光芒社，（1999），http://cruel.org/freeware/cathedral.html）．
12）Sandred, J.: Managing Open Source Project, John Wiley & Sons, 2001（でびあんぐる監訳：オープンソースプロジェクトの管理と運営，オーム社（2001））．
13）SourceForge:http://sourceforge.net とその日本語サイト http://sourceforge.jp.
14）Torvalds, L. and Diamond, D.: Just for Fun: The Story of an Accidental Revolutionary, Harper Business（2001）．（風見　潤訳：それがぼくには楽しかったから，小学館（2001））．
15）オープンソース調査委員会：オープンソースビジネスの動向調査報告書，（社）情報サービス産業協会，13-J010（Mar. 2002）．

（平成 14 年 11 月 4 日受付）