

Title	オープンソース・エコシステムにおける協創の構造 - Eclipseのケーススタディ -
Author(s)	水島, 和憲
Citation	
Issue Date	2009-09
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/8349
Rights	
Description	Supervisor: 井川康夫教授, 知識科学研究科, 修士

修 士 論 文

オープンソース・エコシステムにおける協創の構造
－ Eclipse のケーススタディ －

北陸先端科学技術大学院大学
知識科学研究科知識社会システム学専攻

水島 和憲

2009 年 9 月

修士論文

オープンソース・エコシステムにおける協創の構造
－ Eclipse のケーススタディ －

指導教官 井川 康夫 教授

北陸先端科学技術大学院大学
知識科学研究科知識社会システム学専攻

0750605 水島 和憲

審査委員 井川 康夫 教授 (主査)
梅本 勝博 教授
小坂 満隆 教授
日高 一義 教授

2009 年 8 月

“If you build it, he will come.” 「それを作れば、彼が来る」

映画 フィールドオブドリームス より

目次

第1章	はじめに	1
1.1	研究の背景	1
1.1.1	エコシステムの時代	1
1.1.2	オープンソース	1
1.1.3	Eclipse とエコシステム	2
1.2	研究の目的とリサーチクエスチョン	3
1.3	研究方法	4
1.4	本稿の構成	4
第2章	先行研究調査	5
2.1	ビジネス・エコシステム研究	5
2.2	クラスター、バリュー・ネットワークとの違い	7
2.2.1	クラスター	7
2.2.2	バリュー・ネットワーク	7
2.2.3	ビジネスエコシステムとの違い	8
2.3	オープンイノベーション、オープンビジネスモデルとの違い	9
2.3.1	オープンイノベーション	9
2.3.2	オープンビジネスモデル	10
2.3.3	ビジネスエコシステムとの違い	11
2.4	オープンソース研究	11
2.5	本研究の特異性	14
第3章	Eclipse エコシステムの背景	15
3.1	概要	15
3.2	J2EE 対 .NET	17
3.2.1	Java とは、.NET とは	17
3.2.2	.NET の足音	18
3.2.3	Sun の Java 戦略の問題点	18
3.3	IBM の戦略転換	20
3.3.1	ネットワーク・コンピューティング時代	21
3.3.2	ソリューションカンパニー化	22
3.4	IBM の勝算	23

3.4.1	IBM PC 互換機市場の創出	23
3.4.2	Linux オープンソース活動への参加	24
3.4.3	オープンな市場なら勝てる	25
3.5	まとめ	26
第4章	Eclipse エコシステムの進化の過程	28
4.1	概要	28
4.2	IBM PC エコシステム成功要因	28
4.2.1	オープンな仕様とプラグインアーキテクチャ	28
4.2.2	プラットフォームリーダーの交代	29
4.2.3	エコシステム成功因子	30
4.3	1998年: 開発ツールとしてのプラットフォーム	31
4.4	2001年: Eclipse 誕生	33
4.4.1	オープンソース化	33
4.4.2	Eclipse コンソーシアム	33
4.5	2004年: Eclipse の改革	34
4.5.1	誤算と原因	34
4.5.2	改革内容	36
4.6	2006年: 定期リリース	39
4.7	まとめ	39
第5章	インタビュー	41
5.1	概要	41
5.2	インタビュー調査:参加の理由	41
5.3	Eclipse 市場が理由	41
5.4	開発コスト削減が理由	44
5.4.1	コミュニティ育成支援サービス	45
5.5	IP 管理プロセスが理由	48
5.5.1	IP 管理プロセス:クリアランス手続き	49
5.6	EPL ライセンスが理由	50
5.6.1	EPL ライセンスと価値共有	51
5.7	まとめ	53
第6章	オープンソースエコシステムにおける協創の構造 (モデルの提示)	55
6.1	概要	55
6.2	Eclipse のアーキテクチャ	55
6.3	協創のプロセス	56
6.4	量的な発展プロセス	57
6.5	質的な発展プロセス	58

6.6	まとめ	59
第7章	オープンソース・エコシステムの課題	60
7.1	オープンソースであることによる協創の限界	60
7.2	Eclipse エコシステムの末端における協創の限界	62
7.3	量的な発展プロセス不全	63
7.4	まとめ	64
第8章	結論	65
8.1	本研究によって得られた新たな知見	65
8.2	理論的含意	66
8.3	実務的含意	67
8.4	今後の研究課題	67
付録A	インタビュー	71
A.1	インタビュー概要	71
A.2	インタビュー結果	71
A.2.1	Genuitec	71
A.2.2	Innovent Solutions	72
A.2.3	AvantSoft	72
A.2.4	Sopera	72
A.2.5	Webtide	73
A.2.6	Protecode	74
A.2.7	Meristic	74
A.2.8	Actuate	75
A.2.9	Eclipse 財団	75
A.3	まとめ	76
付録B	Eclipse 参加ベンダーの変遷	77

目 次

1.1	Eclipse のスナップショット	3
1.2	リサーチクエスチョン	4
2.1	オープンソースとビジネスの変遷	13
3.1	Gartner による Java と .NET のシェア予測 (Driver (2002))	19
3.2	Eclipse のマイクロ経済的構造	27
4.1	開発ツールの共通機能のプラットフォーム化	31
4.2	Eclipse プラットフォームのアーキテクチャ	32
4.3	Java 開発ツールの使用状況 2002 年 (上段) と 2003 年 (下段)(小柴 (2003) より引用)	35
4.4	Eclipse のメンバー数の変化 (付録 B より集計)	36
5.1	Eclipse ダウンロード数の推移 (Eclipse Foundation (2009) より引用)	42
5.2	地域ごとのダウンロード数の推移 (Eclipse Foundation (2009) より引用)	43
5.3	EPL ライセンスの権利と義務	53
6.1	技術と財団からみる Eclipse のアーキテクチャ	56
6.2	オープンソースエコシステム協創プロセス	57
7.1	EPL ライセンスによる木構造	63

表 目 次

2.1	ビジネス・エコシステムにおけるネットワーク戦略の分類 (イアンシティ, レビン (2007) より引用)	6
2.2	クラスター、バリュー・ネットワーク、ビジネス・エコシステムの比較 (Peltoniemi and Researcher (2004))	8
2.3	クローズドイノベーションとオープンイノベーションの比較 (チェスプロウ (2004) から引用)	10
2.4	オープンイノベーション・オープンビジネスモデルとエコシステムの違い	12
2.5	オープンイノベーションとしてのオープンソース戦略 (West and Gallagher (2006) より引用)	14
3.1	Eclipse と .NET の開発年表	16
4.1	IBM PC と Eclipse エコシステムの施策	40
5.1	Eclipse のインキュベーションプロセス (Duenas, G., Cuadrado, Santillen and Ruiz (2007) からの引用)	47
5.2	知的生産物としてのソフトウェアとライセンス	51
5.3	ベンダーが Eclipse エコシステムに参加する理由	53
7.1	ビジネス化されているオープンソース (West and Gallagher (2006) から引用)	61

第1章 はじめに

1.1 研究の背景

1.1.1 エコシステムの時代

近年、エコシステムという言葉が頻繁にメディアに登場するようになった。この背景には、エコシステムの成功がビジネスの成功を意味するようになったことがある。つまり、単に、製品やサービスを提供するだけではなく、それらを核として強固なエコシステムを構築したものだけが、ビジネスを成功に導くことができるようになった。例えば、Apple社 iPhone の成功の背景には、製品としての完成度の高さもさることながら、開発者のインセンティブを設け iPhone 用のアプリ開発を推進させるなどして、iPhone を取り巻く関連事業が全体として活性化するエコシステムを構築できたこともある。

以前は、製品やサービス単位での競争であったが、現代は既にエコシステムの競争の時代に入っている。iPhone の例であきらかなように、エコシステムの勝利は一人勝ちに直結する。製品やサービスで競争していた時代は、市場が既に存在することが大前提であり、その中でのシェア獲得をめぐる群雄割拠していた。一方、エコシステムは、製品やサービスを核として、参加者(多種多様なベンダーおよび顧客も含まれる)を増やししながら、すなわち市場を創出しながら、発展していく。つまり、エコシステムは市場創出を行いながら競争するため、エコシステムの覇者が市場の覇者となる。

エコシステムの内部では、多種多様なステークホルダー(ベンダーや顧客)が複雑に絡み合いながら、競争と協調を繰り返し、イノベーションやビジネスモデルが創出される。エコシステムのマネジメントにおいては、内部の競争と協調を促進しつつ、外部との競争には勝利しなければならず、微妙なバランスが要求される。

以上のように、現代はエコシステムの構築がビジネス上の重要な位置を占めつつある時代になった。

1.1.2 オープンソース

ソフトウェアの世界では、オープンソースと呼ばれるソフトウェア開発活動が注目されるようになって久しい。

オープンソースでは、ソフトウェアの知的財産であるソースコードを公共財として公開し、世界中の開発者が協調し合いながらソフトウェア開発を行う。元々、オープンソース

活動は研究者や開発者のボランティア活動をベースとして自然発生的に始まった。初期のオープンソースコミュニティは、あまりビジネス色はなく、技術者の純粋な興味で運営されていた。

ところが、Linux の登場により、ボランティアベースで開発されたオペレーティングシステムが商用製品に匹敵する機能と品質を持つことに、誰もが驚いた。そして、OSS の周辺にビジネスを展開する商用ベンダーが登場した。ベンダーや政府は、オープンソースのパワーをうまく活用して、ビジネスや経済を活性化したいと考え、さまざまな取り組みを行っている。例えば、日本では経済産業省系の独立行政法人情報処理推進機構 (IPA)¹ が OSS 利用事業に対して資金援助を行っている。

今日のソフトウェアベンダーにとって、オープンソースをいかにうまく活用するかが課題になっている。

1.1.3 Eclipse とエコシステム

Eclipse は、図 1.1 に示すような、オープンソースの統合ソフトウェア開発環境である。誰でも無償で、入手、改変、再配布できる。Java の開発者を中心に広く普及しており、Java 開発ではデファクトスタンダードとなっている。また、Eclipse は Java 用の単なる開発ツールではなく、開発ツールのプラットフォームを志向している。そのための仕掛けが用意されており、新たな機能はプラグインという形で柔軟に追加することができる。プラグインとして、テストツールや C 言語や PHP 言語など他のプログラミング言語用のものも存在する。

Eclipse は、2001 年に、IBM が 4000 万ドルの資産価値があるとされる開発環境をオープンソース化したことが始まる。IBM は、当初からエコシステムの構築を謳っていた。

筆者が、Eclipse を知ったのは、2002 年の夏、Eclipse 2.0 がリリースされたときであった。早速、Eclipse をダウンロードし試用したところ、その完成度の高さに驚き、ソフトウェアの開発現場で利用して、積極的に周りに啓蒙していった。また、2002 年 10 月には、Eclipse についての情報共有のための Wiki サイト、エクリップス²を公開した。すると、サイトへのアクセス数がすさまじい勢いで増えていくのを目の当たりにした。1 年後の 2003 年 9 月の時点では、1 日あたり 3,000 人のサイト訪問者、約 16,000 ページビューにまで成長した。

そして、素朴な疑問が沸き上がった。IBM はなぜこのように優れたソフトウェアを無償のオープンソース化したのだろうか。これが、本研究の原点である

¹<http://www.ipa.go.jp/>

²<http://eclipsewiki.net/>

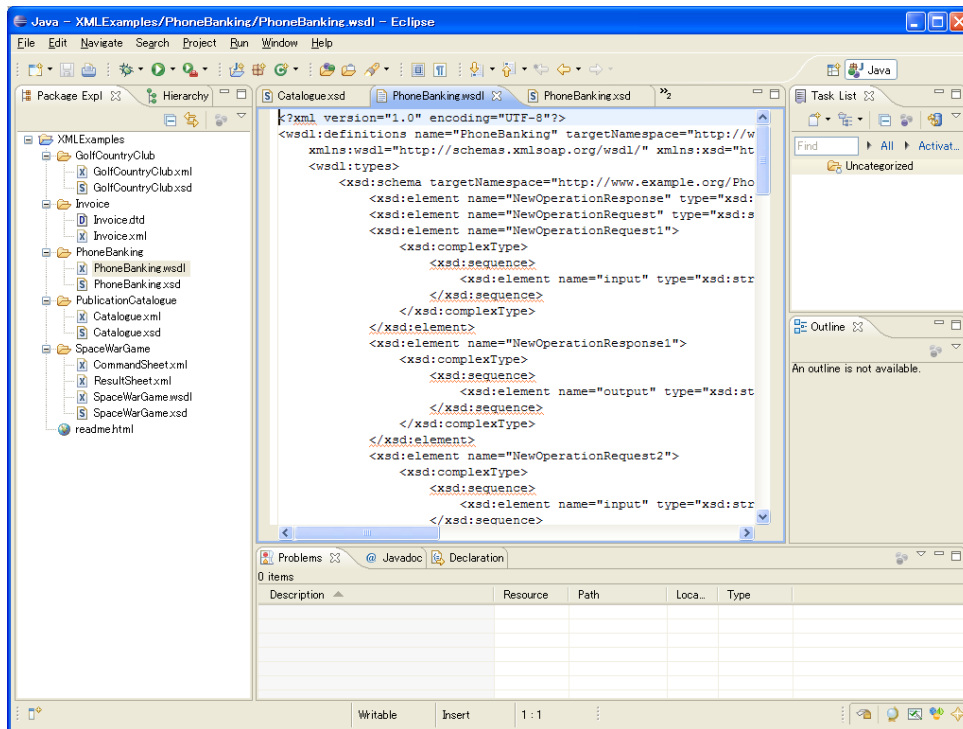


図 1.1: Eclipse のスナップショット

1.2 研究の目的とリサーチクエスチョン

本研究の目的は、Eclipse をひとつのケーススタディとして、オープンソースエコシステムにおける協創の構造を明らかにすることである。Eclipse エコシステムの誕生と成長過程を追いながら、参加企業へのインタビューを通して、エコシステムの複雑な協創の構造に迫る。

リサーチクエスチョンは以下ようになる。

MRQ Eclipse エコシステムにおける協創の構造は何か

SRQ1 IBM は Eclipse のオープンソース化をなぜ始めたのか

SRQ2 どのように Eclipse のエコシステムを構築したのか

SRQ3 Eclipse エコシステムに参加するのはなぜか

図 1.2 に示すように、各 SRQ はそれぞれ、

SRQ1 エコシステムと外部との関係

SRQ2 エコシステムのメカニズム

SRQ3 エコシステムの魅力

を明らかにするための問いである。

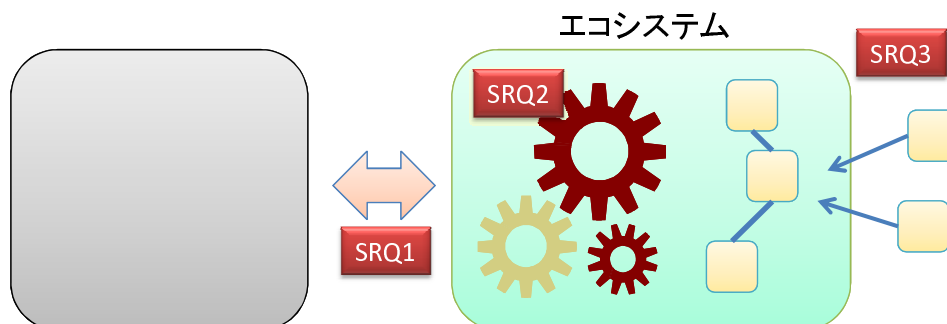


図 1.2: リサーチクエスチョン

本研究の理論的含意は、これまであまり明らかにされてきていなかったエコシステムの誕生と成長過程、成長を促す構造について明らかにすることにある。

実務的含意は、ビジネス・エコシステムの重要性が増している現在、本研究の成果が、エコシステムを構築しようとしている企業、エコシステムに参加しようとしている企業に、意味のある示唆を与えることである。

1.3 研究方法

Eclipse が誕生した 2001 年当時のソフトウェア業界の状況を当時の調査報告や記事から背景を明らかにする。Eclipse のメカニズムについては、Eclipse の変化を議事録やライセンス、会則の履歴を仔細に追いかけて、また eclipse.org で公開されている情報を調査した。Eclipse の魅力については、参加ベンダーへの直接のインタビューを元に参加の動機についても調査した。

1.4 本稿の構成

本稿の構成を以下に示す。

2 章では、エコシステムに関連する先行研究を紹介し、本研究の特異性を説明する。3 章では、SRQ1 のなぜ IBM が Eclipse を始めたのかについて論ずる。4 章では、Eclipse エコシステムの進化の過程を追いながら、エコシステムの仕掛けについて明らかにする。5 章では、4 章で明らかになった仕掛けのうち、ベンダーは何に惹かれて参加するのかを明らかにする。そして、6 章では、オープンソースエコシステムの協創プロセスモデルを提案する。7 章でオープンソースエコシステムの構造が引き起こす課題を列挙し、最後の 8 章でまとめる。

第2章 先行研究調査

2.1 ビジネス・エコシステム研究

経営戦略や組織間関係を説明するために、最初にエコシステムという概念を導入したのは Moore (1993, 1997, 2005) である。企業間の関係に従来とは異なって競争と協創とが同時に併存する現象を認めて、この現象に対して元々は生態学の用語であったエコシステムという言葉を用いて解説した。ビジネスエコシステムは Moore (1997) の中で次のように定義されている。

ビジネス・エコシステムとは、組織や個人すなわちビジネスの世界の有機体どうしの対話を土台とした経済的コミュニティである。この経済的コミュニティは、価値のある製品とサービスを顧客に提供する。そして、顧客自身もエコシステムのメンバーである。エコシステムのメンバーには他に、サプライヤー、製品リーダー、競合およびその他のステークホルダーが含まれる。時間と共に、メンバーは能力と役割を共進化させ、1社もしくは数社の中核企業が設定した進路に自らを合わせるようになる。中核企業が持つリーダーシップの役割には時間と共に変化するかもしれないが、エコシステムのリーダーという機能はコミュニティによって価値が高まる。なぜなら、メンバーが、投資を合わせたり、相互補完的な役割を見つたりするための共有されたビジョンに向かわせるからである。

また、次のように変化してきていることが、ビジネスエコシステムの背景にあると指摘している。

- 競争が、効率や効果ではなく、継続的なイノベーションに移ったこと
- 企業は、企業の現在の製品やサービスなど見えるアセットではなく、イノベーションの軌跡によって定義されるようになったこと
- どの企業も単独ですべてのを行うスキルもリソースもなく、イノベーションを共進化させる組織が必要になったこと

Iansiti and Levien (2004); イアンシティ, レビーン (2007) は、生態学やネットワーク理論の知見を用いてビジネスエコシステムを捉え直した。ビジネス・エコシステムにおいて観察される役割を分類し、さらにエコシステムの健全性という指標を提案している。

Iansiti and Levien (2004) イアンシティ, レビーン (2007) は、ビジネスエコシステムにおけるネットワーク戦略を4つに分類し、それぞれの戦略様式を整理している(表 2.1)。

表 2.1: ビジネス・エコシステムにおけるネットワーク戦略の分類(イアンシティ, レビーン (2007) より引用)

戦略	定義	存在	価値創出	価値獲得	主な焦点と課題
キーストーン	エコシステム全体の健全性を積極的に改善し、自社の持続的なパフォーマンスにも便益を享受する	影響力は大きい、物理的な存在感は一般的に小さい。比較的少数のノードのみを占有する	価値創出の結果の大半をネットワークに残しておく。自社内で創出した価値も広く共有する	ネットワーク全体で価値を共有する。特定の領域では、価値の獲得と共有のバランスをとる	プラットフォームを創出し、ネットワークにおける問題の解決方法を共有する。重要な課題は価値の獲得と共有のバランスをとりながら、価値創出を持続させること。どの領域を選択して占有するかという決定も、重要な課題である
支配者	垂直的あるいは水平的に統合し、ネットワークの大部分をコントロールする	物理的な存在感が大きい。大半のノードを占有する	価値創出の活動の大半を単独で行う	価値の大半を自社のみで独占する	コントロールと支配権を追求する。ネットワークが何を行うかを決定し、直接指示する
ハブの領主	ネットワークをコントロールはしないが、できるだけ多くの価値を横奪する	物理的な存在感は小さい。ごく少数のノードのみを占有する	価値創出はネットワークの他のメンバーに依存する	価値の大半を自社のみで独占する	根本的に整合性のない戦略。領主は価値の源泉としてネットワークに依存しながらも、ネットワークを拒否する。領主は同時に、価値の大半を横奪しており、自らの存在をリスクにさらしている
ニッチ・プレイヤー	自社をネットワークの他の会社と差別化するための特殊な能力を開発する	個々には極めて小規模な物理的存在感。しかしニッチのかたまりとしてはエコシステムの多くの拠点を占める	健全なエコシステムの価値の大半を集合的に創出する	自ら創出した価値を獲得する	キーストーンによって提供されるサービスを利用しながら、自らが能力を有するあるいは開発できる領域に特化する

また、エコシステムでは「価値創出と価値獲得を注意深くバランスさせることが必要である」として、次のようにビジネスエコシステムの健全性を評価するための3つの指標を提案している。

生産性 イノベーションを低コストで、製品や機能に変換できるかどうか。ただし、生産性は持続的に改善されなければならない。

堅牢性 外部環境・状況が変化してもエコシステムへの参加者およびエコシステム自身は生存し続けることができるか。

ニッチ創出 長期間にわたって意味のある多様性を生み出す役割を持ち、新しく貴重な機能を創出できているか。企業の多様性の増大と製品および技術の多様性の増大という2つの観点がある。

最近の研究では、Iansiti and Richards (2006) は、エコシステムの健全性を実際の IT 関連のプロダクト、ブラウザや Web サーバなどのデータを用いて分析している。また、Adomavicius, Bockstedt, Gupta and Kauffman (2006) は、エコシステムの技術進化のモデルを提唱し、デジタル音楽について技術進化のパターンを分析している。

2.2 クラスタ、バリュー・ネットワークとの違い

ビジネスエコシステムとクラスタ、バリューネットワークとの違いについて、解説する。この節は主に、Peltoniemi and Researcher (2004) に依拠する。

2.2.1 クラスタ

米国の経営学者マイケル・E・ポーターが提示した概念で、「特定分野における関連企業、専門性の高い供給業者、サービス提供者、関連業界に属する企業、関連機関（大学、規格団体、業界団体など）が地理的に集中し、競争しつつ同時に協力している状態」をいう（ポーター (1999)）。クラスタに属する企業は、ひとつの街もしくはひとつの地域に集中することがある。

ポーターによればクラスタの利点は、その内部で起こる激しい競争にある。競争によって、企業が切磋琢磨し、各企業の能力が向上する。企業は、基本的に競争関係にあるため、ニーズや技法、技術についての情報が交換されることはあまりない。労働者間の非公式な交流によって、情報がバイヤーやサプライヤーおよび関連する産業分野の間で交換されるくらいである。また、クラスタにおいては、大学が知識の提供元として重要な役割を果たすことがある。大学が知識の提供元となり、企業が知識の消費者となる。逆はない。

2.2.2 バリュー・ネットワーク

ある共通するニーズを持つ顧客層と、それに価値を提供する企業群によって構成される機能的な集合体のことである。企業の投資判断を束縛する収益・コスト構造や価値基準は企業自身の意志や判断ではなく、企業の外部環境によって定まることを説明する概念として、リチャード・S・ローゼンブルーム (Richard S. Rosenbloom) とクレイトン・M・クリステンセン (Clayton M. Christensen) が導入した (クリステンセン (2001))。

バリュー・ネットワークにおいては、ネットワークに属する企業どうしが対話しながら協調する。企業がバリュー・ネットワークに参加する動機は、売上を上げて、コストを下げることにある。顧客のニーズがあり、その周りに企業が群がり活動する構図になっている。ネットワークの内部では、ユニークなコンピテンシーを持つ企業が他のメンバーから選択される。バリュー・ネットワークは地域に縛りつけられない。グローバルに広がっていることもある。

バリュー・ネットワークにおいては、注文数、供給量、品質情報などの運営上の情報は流通するが、イノベーションを起こすような知識は流通しにくい。

2.2.3 ビジネスエコシステムとの違い

クラスター、バリューネットワークとビジネスエコシステムとの違いを表 2.2 にまとめる。

表 2.2: クラスタ、バリュー・ネットワーク、ビジネス・エコシステムの比較 (Peltoniemi and Researcher (2004))

	クラスター	バリュー・ネットワーク	ビジネス・エコシステム
地理	地理的に集中	ローカルからグローバルまですべて	地理的な差異は意味がない
競争と協調	激しい競争	協調	競争と協調が同時
産業	同じ産業の企業	互いに補完し合う異なる産業	産業という概念では捉えられない。ビジネスエコシステムと呼ぶべき
知識創造・移転	競争しているので積極的な共有は限定的	運営情報に限定	相互に連携しているため知識創造と移転が可能。共有された運命が知識共有と共同的な知識創造の動機
制御主体	メンバーは一様に独立	ひとりの強力なアクター	分散意思決定

クラスターやバリューネットワークでは、コミュニティ構成の要素として地理的な違いに意味があるが、ビジネスエコシステムの時代においては、インターネットに代表される最新のコミュニケーション技術とグローバルな競争によって地理的な重要性がなくなってしまっている。

クラスター内部では激しい競争があり、一方、バリューネットワーク内部では各企業は協調して活動している。これに対し、エコシステムの内部は競争と協調が同時に起こっている。

クラスターは同じ産業に属する企業の集合体である。バリューネットワークは互いに補完し合う異なる産業が連携し合う。Moore は、競争と協調が同時に起こっているのもはや産業という概念では捉えられないとして、代わりにビジネスエコシステムと呼んだ。

知識の創造と移転については、クラスターでは基本的に競争しているため積極的な知識の共有は限定的である。非公式な人の行き来により共有される程度である。バリューネットワークでは、注文数や供給量などの運営上の情報が流通するだけで、知識の流通はほとんどない。ビジネスエコシステムでは、知識創造と知識移転は、企業が相互に連携しているため可能であり、運命共同体であることがそれらのモチベーションとなっている。

クラスターでは、メンバーは一様に独立しているため、制御は必要ない。バリューネットワークでは、ひとつの強力なアクターがいる。そのため、その他多くの小さなサプライヤーは強力なアクターに依存することになる。エコシステムにおける制御は、分散的である。キーストーンとよばれるアクターがネットワークの中心的な役割を果たすことがあるが、バリューネットワークのような統制力はない。

2.3 オープンイノベーション、オープンビジネスモデルとの違い

オープンイノベーション、オープンビジネスモデルとの関係を整理する。

2.3.1 オープンイノベーション

オープンイノベーションはヘンリーチェスブロウが「OPEN INNOVATION ハーバード流イノベーション戦略のすべて」(チェスブロウ (2004))の中で提唱した概念である。著書の中で、クローズドイノベーションが時代にそぐわなくなり、新しいアプローチの出現を主張している。そして、その新しいアプローチに「オープンイノベーション」と名付けた。

企業が技術革新を続けるためには、企業内部のアイデアと外部(他社)のアイデアを用い、企業内部または外部において発展させ商品化を行う必要がある。オープンイノベーションは、企業内部と外部のアイデアを有機的に結合させ、価値を創造することをいう。オープンイノベーションは、アイデアを商品化するのに、既存の企業以外のチャンネルをも通してマーケットにアクセスし、付加価値を創造する。(P8)

これまでは、企業の内部で生まれたアイデアが研究開発プロセスの途中で、たとえ有望であったとしても企業のビジネスにマッチしなければ捨て去られてしまうことがあった。オープンイノベーションでは、単に捨ててしまうのではなく、外部に技術供与したり、逆

に外部で生まれたアイデアを企業内部に取り込んだりして、アイデアを有効に活用し価値創造を行う。

クローズドイノベーションとオープンイノベーションの比較表を表 2.3 に示す。

表 2.3: クローズドイノベーションとオープンイノベーションの比較 (チェスブロウ (2004) から引用)

クローズドイノベーション	オープンイノベーション
最も優秀な人材を雇うべきである	社内に優秀な人材は必ずしも必要ない。社内に限らず社外の優秀な人材と共同して働けばよい。
研究開発から利益を得るためには、発見、開発、商品化まで独力で行わなければならない	外部の研究開発によっても大きな価値が創造できる。社内の研究開発はその価値の一部を確保するために必要である
独力で発明すれば、一番にマーケットに出すことができる	利益を得るためには、必ずしも基礎から研究開発を行う必要はない
イノベーションを初めにマーケットに出した企業が成功する	優れたビジネスモデルを構築する方が、製品をマーケットに最初に出すよりも重要である
業界でベストのアイデアを創造したものが勝つ	社内と社外のアイデアを最も有効に活用できた者が勝つ
知的財産権をコントロールし他社を排除すべきである	他社に知的財産権を使用させることにより利益を得たり、他社の知的財産権を購入することにより自社のビジネスモデルを発展させることも考えるべきである

2.3.2 オープンビジネスモデル

オープンイノベーションでは、テクノロジーに力点が置かれていた。その後、チェスブロウはオープンにするものをテクノロジーからビジネスモデルへと発展させている。

チェスブロウは、「オープンビジネスモデル 知財競争時代のイノベーション」(チェスブロウ (2007)) の中で、ビジネスモデルをオープンにすべきであると主張している。なぜであろうか。

新興国に追い上げられている先進国が、過去と同様のペースでイノベーションを継続し続けなければならないという。そのためには、「オープンイノベーション」でチェスブロウが主張したように、「イノベーション活動の分割」こそが鍵である。そして、「分割の機

会を追求するためには、企業は自社のビジネスモデルとオープン化する必要がある」。なぜなら、「(ビジネスモデルの) オープン化が実現できれば、より多くのアイデアが検討対象になり、企業内にとどまっていたアイデアが市場に持ち込まれ、停滞していた経済を改善してくれる可能性が増す」からである。ビジネスモデルには「価値を創出すること」と「創出された価値の一部を収穫すること」という2つの重要な機能がある。これが、オープン化によってレバレッジが効く。価値創出については、「オープンなモデルにより、はるかに多くのアイデアを活用して価値を創出できる。社外の多様なアイデアを取り込むことができるようになるからである。」という。また、価値収穫については、「オープンなモデルは、より多くの価値を収穫可能にしてくれる。企業自身のビジネスだけではなく、他社のビジネスの主要な資産・資源・地位を活用することができるようになるからである。」という。

2.3.3 ビジネスエコシステムとの違い

内外のアイデアを組み合わせる価値を創出するという目的においては、エコシステムもオープンイノベーション/オープンビジネスモデルも大きくは変わらない。

違いは視点にあるといえる。オープンイノベーションやオープンビジネスモデルでは視点は常に企業にある。すなわち、オープンイノベーションやオープンビジネスモデルでは、「内」と「外」という概念で解説されることから分かるように、基本的に内(自社)と外(それ以外)という二項関係でイノベーション活動を捉えようとしている。そのため、基本的には自社のコアの技術やビジネスは自社内に留め、自社では活用しきれないものについてはオープンにして他社との協調を模索する。

一方、エコシステムの視点は「環境」もしくは「場」にある。エコシステムでは、「場」をいかに継続発展させていくのみに腐心する。「場」全体のマネジメントに力点があるため、エコシステム内部には競争と協調が同時に発生することを認め、また中核企業が交代することも許容している。「場」を活性化するためには、自社のコアの技術やビジネスモデルをオープンにすることもある。実際、Eclipse エコシステムではコア技術であるプラットフォームをオープンソースとして公開し、それがエコシステムの活性化の誘因となっている。

表 2.4 にまとめる。

2.4 オープンソース研究

本研究の対象である Eclipse は、オープンソースを核としてエコシステムを構築している。オープンソースについては、Linux の成功もあって、先行関連研究は多い。

エリック・レイモンドは、「伽藍とバザール」(レイモンド (1999)) という論文の中で、新旧のオープンソース開発手法を区別し名付けを試みた。エリック・レイモンドは、成功した Linux の開発手法を「バザール方式」と呼び、それ以前の開発手法を「伽藍方式」と

表 2.4: オープンイノベーション・オープンビジネスモデルとエコシステムの違い

項目	オープンイノベーション・オープンビジネスモデル	エコシステム
オープンにするもの	自社では活用できない技術やビジネス	エコシステム全体を活性化する技術やビジネス。自社のコア技術・ビジネスをオープンにすることもある。
競争と協調	協調	競争と協調が同時
視点	自社と他社という二項関係	エコシステム全体 (場、環境)

呼んだ。伽藍方式では、選ばれた比較的少数の開発者がソフトウェア開発に従事し、ある程度まとまった形になるまで外部には公開しない。一方のバザール方式では、開発参加者を特定しない。誰でもが参加できる。また、参加者の独立性を尊重し、中央集権的な管理は行わない。明確なリリースのタイミングは存在せず、開発の初期段階からすべてを公開していく。これによって、さらに多くの参加者を募ることができる。当時、Linux がなぜボランティアベースの活動なのに高品質かつ生産性の高いソフトウェア開発ができるのかが疑問であった。エリック・レイモンドは、自ら、オープンソースソフトウェア fetchmail の開発プロジェクトを立ち上げ、アクションリサーチを行いながら、Linux の秘密を解きあかした。

オープンソース開発では、コミュニティで開発が進められる。スティーブン ウェバーは、「オープンソースの成功」(ウェバー (2007)) でオープンソースプロセスを政治学者の視点で分析を試みている。社会的視点、政治的視点、経済的視点から多層的な説明モデルを構築し、オープンソースプロセスが別の分野にも適用できる可能性があることを示唆している。

オープンソースソフトウェアのベースとなった概念にフリーソフトウェアがある。ただ、フリーソフトウェアは自由なソフトウェアを意味し、もともとプロプラエタリーなソフトウェアに束縛されるのではなく自由になることを目的としていたため、ビジネスとの親和性が悪かった。そこで、エリック・レイモンドらはオープンソースという言葉を編み出し、ソフトウェアのソースコードを公開しつつビジネスを排除しない道を作った。これによって、オープンソースという概念が浸透し、フリーソフトウェアでは敬遠していた企業がビジネス機会を認めてオープンソース活動に参加するようになった。その場合に、オープン化とビジネスとのバランス、すなわち価値提供と価値獲得のバランスが重要になる。佐々木, 北山, 国領 (2000) は、Linux のオープンソースコミュニティとビジネスとがうまく棲み分けながら、相互に発展していく現象を整理している。RedHat を始めとして Linux ではコミュニティが生み出したオープンソースをディストリビューションという形で提供しサポートサービスでビジネスを行っていた。企業は Linux オープンソースの

成長を阻害しないように不干渉の立場を取り、量産される Linux オープンソースを編集してディストリビューションとして提供ことにユーザは価値を見いだしているとしている。Riehle (2007) は、これまでのボランティアベースのものを「コミュニティオープンソース」、企業が主導的に管理するオープンソースを「商用オープンソース」と呼んで区別している。West and Gallagher (2006) には、オープンソースにおけるさまざまなライセンスとビジネスモデルについて整理し、どのように価値提供と価値獲得のバランスをとっているのかをまとめている。

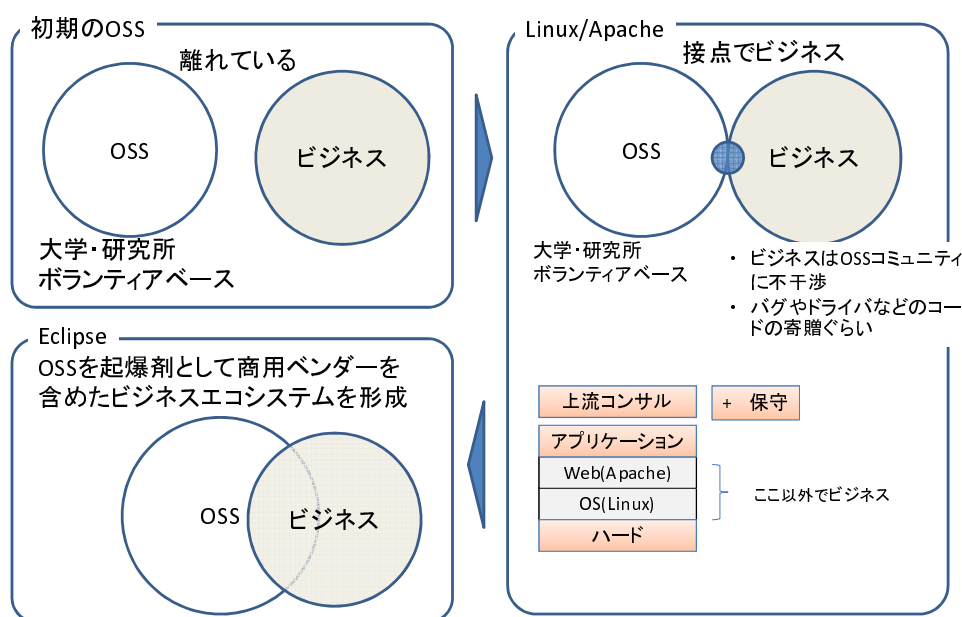


図 2.1: オープンソースとビジネスの変遷

Linux の成功をみて誰もが、オープンソースの市場創出力、イノベーションのパワー、開発生産性の高さ、開発スピードの速さに驚き、自社のソフトウェアをオープンソースによってうまく活性化できないものかと夢見たはずである。

一時期、社内では使用しなくなったソフトウェアや保守しきれなくなったソフトウェアのソースコードをオープンソース化して、公開するという動きがあった。ところが、それらはほとんどが失敗に終わった。単にソースコードを公開しただけでは、オープンソースコミュニティが立ち上がらなかった。

日本でも、いくつかの試みがあった。その中でも野心的な試みが「セルベッサ」(湯澤 (2005) 竹田, 米山 (2002)) である。セルベッサは、外食チェーン大手のニュートーキョーが食材発注システムとして自ら開発した業務アプリケーションをオープンソースとして1999年11月に公開した。その後、2000年に、資本関係のない競合3社(アウトバックステーキハウス、カプリチョーザ、大戸屋)が採用、2001年ダイナック、2002年名古屋の浜木綿、2003年R&D外食ネットで順次稼働したが、2005年の時点ですでにアウトバックステー

キハウス、カプリチャー、大戸屋では利用を停止しているなど、オープンソースとして成功しているとは言い難い。

このように、ソースコード公開してオープンソースのパワーをうまく活用したいと考えても、なかなか運用がうまくいかない。ところが、Eclipse エコシステムの傘下では多くのオープンソースプロジェクトが生まれ、全体として活発に活動しており成功している。

表 2.5: オープンイノベーションとしてのオープンソース戦略 (West and Gallagher (2006) より引用)

オープンソース戦略	例	内部イノベーションの最大化	外部イノベーションの役割	外部イノベーションの動機付け
R&Dのプール/ 製品開発	Linux	共同参加し成果を共有	貢献をプールし全員が利用	推進中の機関が正統性と継続を構築
スピアウト	Eclipse	他のゴールを支援するため非商用技術の種まき	内部イノベーションを進行中のイノベーションに取り替える	価値のある技術に無償でアクセス可能
補完財の販売	Apache	製品全体の上位の価値をターゲットとする	外部のコポーネントを内部開発の基礎として活用	企業はコポーネントの継続的供給元として結合
コンポーネントの寄贈	Half-Life	外部コントリビュータに拡張可能なプラットフォームを提供	製品構築の多様性と新味を追加	表彰と金銭以外の報奨

2.5 本研究の特異性

Moore (1993, 1997); イアンシティ, レビーン (2007) 等では、ウォールマートやインテル、Microsoftなどをエコシステムであるとして研究対象に上げている。これらのエコシステムはいずれも、偶然の産物として分析されている。ところが、Eclipse エコシステムは、意図的に構築されたエコシステムであり、エコシステムを形成するためのさまざまな仕掛けやバックグラウンドがある。また、オープンソースのパワーをうまくエコシステムに活用しているという点でも、Eclipse エコシステムの試みは挑戦的である。本研究は、意図的に形成された Eclipse エコシステムに潜む構造を解きほぐしながら、知識の集約と創出、ビジネスの競争と協調のマネジメントについて論ずる。また、Eclipse エコシステムにおける協創の構造が内包する課題についても考察する。

第3章 Eclipseエコシステムの背景

3.1 概要

2001年11月5日にIBMはアプリケーション開発のプラットフォームをオープンソースコミュニティ(Eclipseコンソーシアム)に寄贈した。このソフトウェアは、元々IBMの次世代WebSphere Studio製品の基盤としてOTI¹と共同開発してきたツールであり、4000万ドル相当の資産価値があるといわれた。それを無償のオープンソースとして提供するという、それまでのIBMとは明らかに異なる行動に対して、多くのメディアは複雑な驚きをもって報道した(Junnarkar (2001))。

この章では、「IBMが2001年になぜ4000万ドルもの資産価値のある開発ツールを無償のオープンソースとして寄贈したのか」というリサーチクエスチョンに対して、当時のソフトウェア業界の状況、IBM PCでの経験、IBMのビジネス戦略の転換、という側面から論理展開する。

本章の研究にあたっては、以下の資料を調査し、遠因となった2001年以前の経緯と直接の原因となった2001年の状況を明らかにする。

- 関連記事: 当時の事情を伝える記事やインタビュー記事
- 調査会社のレポート
- 関連書籍

3.2節では、2001年当時のSunを中心としたJava陣営とMicrosoftの.NET戦略の状況について解説する。JavaのJ2EEと.NETのASP.NETはエンタープライズアプリケーションの開発・実行環境という意味で同じ市場をターゲットとしており、互いに代替財の関係にある。その当時、それまでJava(J2EE)の独壇場であったエンタープライズ向けアプリケーション市場に対して、Microsoftは豊富な資金力をバックに.NETで大攻勢を仕掛けてきていた。これが、直接のきっかけとなったことを示す。

3.3節では、ガースナーのIBMをソリューションカンパニー化させるという戦略の転換があったがために、Eclipseのオープンソース化に踏み切れたのだという推測を展開する。Eclipseのエコシステムの戦略が、ガースナーが改革した戦略に適合していたことを示す。

3.4節では、IBM PCとLinuxオープンソースでの経験があったために、IBMにはオープンソース化することに対して勝算があったことを解説する。

¹Object Technology Internationalの略。URLは<http://www.oti.com/>。IBMが1996年に買収。

表 3.1: Eclipse と .NET の開発年表

年	月	Eclipse	.NET
1998	11	後に Eclipse と呼ばれるようになる Java IDE の開発作業が OTI と IBM の内部で始まる	
2000	6	Eclipse Tech レビュー版リリース	
	7		.NET プレ-ベータ版
	9		.NET 1.0 ベータ 1
2001	3	http://www.eclipsecorner.org/ オープン	
	6	Eclipse 0.9 リリース	.NET 1.0 ベータ 2
	10	Eclipse 1.0 リリース	
	11	IBM が Eclipse にソースコードを寄贈 eclipse.org 発表	
2002	1		.NET 1.0 RTM ²
	6	Eclipse 2.0 リリース	
2003	3	Eclipse 2.1 リリース	
	4		.NET 1.1 RTM
2004	2	IBM から独立し、Eclipse を非営利組織に再編成 Eclipse カンファレンス (EclipseCon2004) 開催	
	6	Eclipse3.0 リリース	
	12	パルミサーノ・レポート「イノベート・アメリカ」を発表	
2005	2	EclipseCon 2005 開催	
	6	Eclipse3.1 リリース	
	11		.NET 2.0 RTM
2006	3	EclipseCon 2006	
	6	Eclipse 3.2 リリース	
	11		.NET 3.0 RTM
2007	3	EclipseCon 2007	
	6	Eclipse 3.3 リリース	
	11		.NET 3.5 RTM
2008	3	EclipseCon 2008	
	6	Eclipse 3.4 リリース	
	11		3.5 SP1
2009	3	EclipseCon 2009	
	5		.NET 4 Beta 1
	6	Eclipse 3.5 リリース	

3.2 J2EE 対 .NET

この節では、リサーチクエスチョンの内、次の疑問について分析を試みる。

- なぜ、2001年であったのか?
- なぜ、開発環境 (IDE Integrated Development Environment) であったのか?

3.2.1 Java とは、.NET とは

Java は、Sun が開発したプログラミング言語であり、“Write once, run anywhere” を謳い、特定の OS やマイクロプロセッサに依存せず、基本的にどのようなプラットフォームでも動作することを信条としている。構文は C 言語および C++ 言語から多くを引き継いだオブジェクト指向プログラミング言語であり、ソースコードをコンパイルするとバイトコードに変換される。バイトコードは、JavaVM (Java 仮想マシン) の上で動作する。プラットフォーム毎に JavaVM を用意することによって、一度書いたプログラムをさまざまなプラットフォームで動作させることができるようになっている。

J2EE とは Java 2 Enterprise Edition の略³であり、Sun が中心となって仕様を定めている企業向け業務システムに必要な機能セットを指す。動的なウェブサイトや Web アプリケーション、Web サービスを実現することができる。各ベンダーは、Sun から J2EE のライセンスを受け、ベンダーが個別に J2EE の仕様を実装し販売している。ちなみに、IBM の J2EE 準拠製品は WebSphere と呼ばれる。

.NET とは、Microsoft が提供するネットワークベースのアプリケーションのための開発・実行環境である。.NET の基盤である共通言語基盤 (Common Language Infrastructure, CLI) は、ECMA, ISO, JIS において標準化されており、他のベンダーが独自に実装することもできるようになってはいるが、実質は Microsoft が仕様を定め Windows 向けの実装を行っている。Microsoft の CLI 実装を共通言語ランタイム (Common Language Runtime, CLR) と呼ぶ。CLR は仮想マシンであり、共通中間言語 (Common Intermediate Language, CIL) と呼ばれる、プログラミング言語や環境に依存しない中間言語を解釈・実行する。CLR はちょうど Java の JavaVM に相当するが、Java との違いは .NET ではアプリケーション記述言語を特定していない点である。.NET では既存の C++ 言語や Visual Basic に加えて新たに Microsoft が設計開発した C# 言語も利用可能となっている。なお、C# は公には C/C++ を拡張した言語とされているが、その構文からは Java 言語の影響を多分に受けていることが容易に分かる。

.NET において J2EE に対応するのが ASP.NET である。Web アプリケーションのフレームワークであり、J2EE と同様に動的な Web サイトや Web アプリケーション、Web サービスを実現する。

³2006 年以降は JavaEE (Java Platform, Enterprise Edition) と名称が改められた。本稿では J2EE と Java EE を余り区別せずに使用することにする。

以上みてきたように、Java と .NET は、その設計思想や言語仕様が似ている。ターゲットとしている市場は、ともにエンタープライズ向けアプリケーション市場であり、正面衝突している。すなわち、両者は、ミクロ経済の用語でいうところの互いに代替財の関係にある(クルーグマン, ウェルス (2007))。

3.2.2 .NETの足音

前節で説明したように、Java と .NET, J2EE と ASP.NET は機能およびターゲット市場が非常に似通っており、競合関係にある。

詳しくは3.3節で解説するが、IBMはガースナーの命によりソリューションカンパニー化へのシフトを目指しており、システムのスタックのミドルウェア部分に注力して収益を挙げようとしていた。それが WebSphere という製品であった。その領域に、Microsoft の .NET(ASP.NET) が侵攻しつつあった。

表 3.1 に示すように、.NET Framework のバージョン 1.0 RTM (Release to Manufacturing) がリリースされたのは2002年1月であるが、リリース前の2000年9月にベータ1、2001年6月にベータ2をリリースしていた。.NETは、Microsoft がサーバソフトウェア市場へと進出するための道具として、豊富な資金力を背景に総力を結集して推進していた。Microsoft は、.NET に年間30億ドル超の研究開発費を投入していた(ガワー, クスマノ (2005))。2001年はリリースに向けて、ベータ版を公開しながら徐々にメディアやユーザ企業、サードパーティの関心を集めつつあった時期であった。

複数の調査会社が .NET の成長を予測していた。たとえば、図 3.1 は、Gartner の2002年における市場予測⁴である(Driver (2002))。.NET が大きく伸びることを予測していた。同様に、Meta Group(Sholler (2002)) も、.NET が市場シェアを伸ばして2004年までには30%獲得するであろうと予測していた。Java の市場シェア予測は40%であった。.NET 急進の予測は、Sun や IBM などの Java 陣営にとっては脅威と映っていたはずである。

Vawter and Roman (2001) は、.NET がシェアを大きく伸ばす要因としてその開発環境である Visual Studio .NET を挙げている。Java の開発環境は、.NET の開発環境と比較すると貧弱であるという。なぜ Java の開発環境が弱点となってしまったのだろうか。次節では、その原因が Sun の戦略にあったと論を展開する。

3.2.3 Sun の Java 戦略の問題点

J2EE を含めた Java の仕様は、すべて JCP(Java Community Process) 標準化プロセスを経て決定される。ここで注目すべきは、決定するのは仕様だけであるということである。実装についてはライセンスを受けたベンダーに任されている。しかも、仕様には実装依存部分が残されており、細部については各ベンダーで独自の仕様を定義し実装して良いことになっている。

⁴ただし、このグラフは2002年当時の予測であり、現在のシェアとは異なる

E-Business AD Programming Models

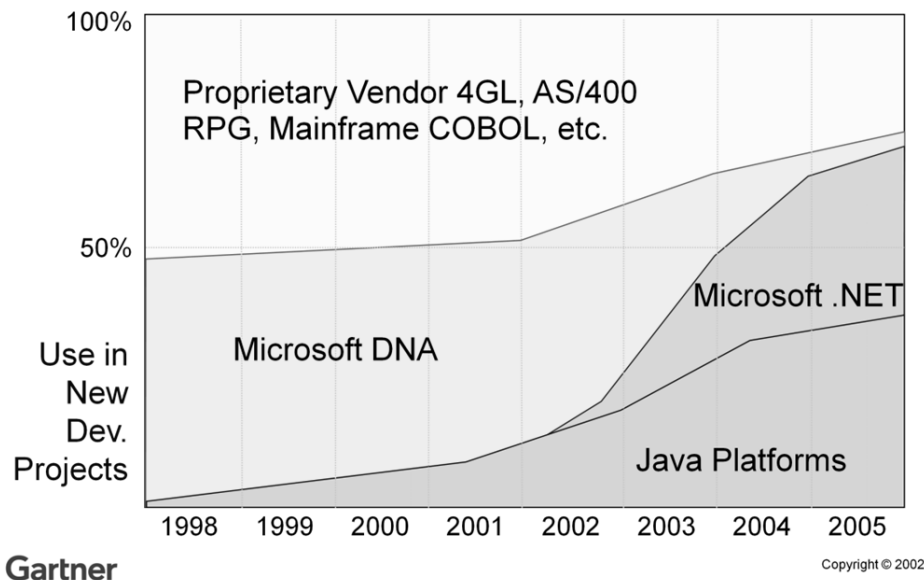


図 3.1: Gartner による Java と .NET のシェア予測 (Driver (2002))

Sun は意図的にベンダー依存部分を残していると思われる。大本の仕様を押さえておき、細部についてはベンダー依存とすることで、Sun がコントロールを効かせながらも、その仕様の元で各ベンダーが競争しながら切磋琢磨し市場を拡大していくということを目論んだのだろうと予測する。基本的に、Sun はハードウェアの会社であるので、ハードウェアよりも上位のミドルウェアの市場が大きくなれば自ずと下位のサーバハードウェアの売上も伸びると踏んだのであろう。この戦略は、競争によって市場が広まるという利点がある一方で、各ベンダーがばらばらになってしまうという状況を生み出してしまっていた。エンタープライズアプリケーションが J2EE だけである場合は、その内部のベンダーどうしの競争になるので問題がないが、エンタープライズ市場に対する外敵が登場するとこの戦略の弱点が露呈する。それが、開発環境であった。

Vawter and Roman (2001) が指摘するように、J2EE と .NET の開発環境を比較した場合、圧倒的に .NET の方が勝っていた。Visual Studio は Windows アプリケーションの開発環境であり、その出来の良さ (操作性および開發生産性) については定評があった。一方、これに対して Java の開発環境については定番の開発環境があったわけではなく、JBuilder などのようにサードパーティ製の開発環境が存在してはいたが高価であったこともあって、多くの Java 開発者は旧態依然とした単純なテキストエディタとコマンドラインによる Java コンパイラの実行・デバッグを行うという有り様であった。当然、開發生産性は低かった。

Vawter and Roman (2001) は、次のように指摘している。

我らの結論は、ツールという点については Microsoft にハッキリと分があるということである。J2EE コミュニティによって提供されるツールセットの機能は全体として、Microsoft によって提供されるツールの機能を上回っている。しかしその一方でそれらのツールは 100 % 相互運用可能ではないのである。なぜならそれらは単一のベンダーから出てくるものではないからである。

当時は、まさに、Java 陣営の開発環境という空隙を突いて、Microsoft が参入しつつあるという状況であった。

以上のように、2001 年はちょうど .NET の台頭を予感させる時期であった。Java 陣営にとっては脅威として映っていたはずであるが、Sun の戦略の元では互いに競合関係にあったために、総力を結集して対抗することが難しかった。特に、開発環境の機能的な陳腐さが弱点であった。

次節では、「なぜ開発環境を無償のオープンソース化したのか」という疑問について、別の視点から論を展開する。

3.3 IBM の戦略転換

.NET の脅威が迫りつつあり、かつ、開発環境に弱点があった。ならば、開発環境を強化して Microsoft の開発環境 Visual Studio と同等もしくはそれ以上の機能を持ったツールを提供し、Microsoft と同じように有償で販売すれば良かったのではなからうか。だが、IBM は有償にはしなかった。4000 万ドルもの投資をした開発環境を無償で提供した。この節では、「なぜ開発環境を無償のオープンソースとしたのか」という疑問の内、

- なぜ開発環境なのか
- なぜ無償なのか

について IBM の戦略の観点から論じ、その理由が、ガースナーが IBM 再建のために取った次の 2 つの大きな賭 (ガースナー (2002)) に強く関係していることを明らかにする。

ガースナーは、IBM 会長に就任したときにメインフレームの凋落が酷く、メインフレームに変わって新たな柱となる戦略を打ち立てる必要があると考えた。そこで、産業の方向性と IBM 自身の戦略について賭をした。

- ネットワーク型コンピューティング・モデルが台頭すること
- IBM をソリューション提供型カンパニーにすること

ガースナーは、

変化が起こるとき、好機をとらえ、動きを主導した企業が飛び抜けて好調になる。そして、それ以外の企業は追随するしかなくなる。(ガースナー (2002))

と判断して、「インターネット革命が起こる前夜」の 1994 年に賭の先手を打った。

3.3.1 ネットワーク・コンピューティング時代

ネットワーク型コンピューティング・モデルとは、「単体のコンピュータがネットワーク型に取って代わられるというもの」である。

この世界が実現すれば、コンピュータの進化の方向も根本的に変化する。ひとつには、この世界がオープンな標準規格に基づくものになるのは確実だと言える。あらゆる企業やユーザー、機器、システムがいつでもどこでも接続できる真のネットワーク社会を実現するには、これ以外に方法はない。こうした標準規格に基づく世界が実現すれば、競争環境は大きく変化する。(ガースナー (2002))

ガースナーは、オープンスタンダードが一般的になるという確信を持ち、競争環境の変化を予感していた。では、どこに競争環境が変化すると考えていたのであろうか。

ガースナーは、ネットワーク型コンピューティング時代のソフトウェア事業でどんな賭をするかを定めるために、ソフトウェアの役割を3段階に分けて考えた。すなわち、最下層に位置するOS、最上層に位置するアプリケーションソフト、そしてその中間に位置し前述の2つを結びつけるミドルウェアである。

最下層は Microsoft が圧倒的な力を持つ OS を所有している。(…) オープン・スタンダードの世界ではさらにありふれた商品になっていくと考えられた。

最上層のアプリケーションソフト市場では、SAP やピープルソフト、JD エドワーズなどの企業が力を持ち、IBM は重要な位置を占めてはいない。

中間では、データベースやシステム管理ソフト、トランザクション管理プログラムなどの製品がある。

(中略)

ネットワーク・コンピューティングに変わったときに重要になるものについて考えを進めていくと、ミドルウェアは取り残された場所ではなく、戦略上の重要な戦場だと思えてきた。(…) ユーザが増える。機器が増える。通信量が増える。アプリケーションやプロセス、システム、ユーザー、企業を統合する方法への需要が高まる。OS はこれらを統合するためのものにはならない。だが、ミドルウェアはまさにそのためのものだ。(ガースナー (2002) P192-193)

ネットワーク・コンピューティングの時代には、「世界がオープンな標準規格に基づくものになることは確実」であると確信し、競争環境が移り「ミドルウェアは戦略上の重要な戦場だ」と考えていた。

こうした理由から、IBM はミドルウェア製品開発に大規模な投資をした。そのひとつが、前述した WebSphere である。WebSphere は、オープンな標準規格である J2EE に準拠したミドルウェアであり、ガースナーの方針にマッチした戦略上重要な製品である。

WebSphere は実行環境である。したがって、その補完財は開発環境となる。ミクロ経済上、開発環境の価格が下がれば実行環境の需要が増える(クルーグマン, ウェルス (2007))。開発環境を無償とした理由のひとつに、この構造があったと推測する。

次節では、開発環境の無償化の理由を別の観点から分析する。

3.3.2 ソリューションカンパニー化

ガースナーはもうひとつ賭をしていた。IBM のソリューションカンパニー化である。

顧客はソリューションを提供できる企業を高く評価するようになる。さまざまなサプライヤーの技術を統合するソリューション、さらに重要なのは、自社の事業プロセスに合わせて技術を統合してくれるソリューションだ。(…) コンピュータ産業は技術主導ではなく、サービス主導になる。(ガースナー (2002) P169)

ガースナーには前職 (RJR ナビスコ CEO) の経験から「顧客が現在の業界構造にしないで我慢できなくなるとの確信があった」。ソリューションカンパニーとは、「顧客向けにコードをいじったりするだけの会社ではない」、「システム構築からアーキテクチャーの決定、コンピュータの管理・運用に至るまで、文字通り顧客の側に立って、すべてを受け、行動する企業」である。

顧客のためのソリューションを提供するためには何が必要であろうか。

顧客に最終的にソリューションとして提供するためには、さまざまな技術をシステムとして統合する必要がある。システム統合には開発が必ず伴う。システムを開発するには開発者が必要になる。折角、ソリューション案件を獲得しても、開発者がいないためにシステム開発ができないのでは話にならない。

では、開発者を増やすためにはどうすればよいだろうか。

いろいろが施策が考えられるが、もし開発生産性の高いツールが無償で提供されており、そのツールをマスターしていれば仕事が見込める場合、多くの開発者が飛びつくのではないだろうか。実際、Eclipse では、多くの Java 開発者がそれまでの貧弱な開発環境から瞬く間に Eclipse に移行した (小柴 (2003))。多くの開発者が同じ開発環境を使うようになると、その使い方を解説するコミュニティサイトや書籍が増え、情報が増えることによって、職場でも詳しい開発者が増えていくという、正の循環、つまりある種のネットワーク外部性が働くようになる。日本でも、EclipseWiki⁵のようなコミュニティサイトが出現したり、Web メディアでの解説や多くの Eclipse 本が出版されたりした。

ソリューションベンダーからみれば、現場に開発者を投入するケースにおいて、同じ開発環境を利用する開発者が多ければ、それぞれ異なる開発環境を使用している場合よりも、比較的スムーズに現場に馴染ませることができる。また、一方、開発者からみても、

⁵<http://www.eclipsewiki.net/> (運営者は著者)

案件ごとに使用する開発環境が異なる場合に発生する学習コストをなくすことができる。デファクトスタンダードの開発環境は、ソフトウェア開発者の流動性を高め、ソリューションベンダーにとっても、システム開発者にとっても、双方にとってメリットとなる。

開発環境を無償提供することによって、開発者が増え、かつ、開発者の流動性が高まる。これによって、ソリューション案件をこなし易くなり、ミドルウェアの売り上げ増につながる。

以上で、ガースナーの二つの賭が開発環境の無償化に関連していることを論じた。

次節では、IBM には勝算があったことを、それまでの IBM の経験から論ずる。

3.4 IBM の勝算

この節では、開発環境の無償オープンソース化について IBM に勝算があったことを、これまでに IBM が経験した次の二つの事柄から論ずる。

- IBM PC 互換機市場の創出
- Linux オープンソース活動への参加

3.4.1 IBM PC 互換機市場の創出

この節では、「オープンな仕様による強力な市場創出力」を IBM が教訓として学習し、これがオープンソース化を発想させたと論理展開する。

IBM PC は 1981 年フロリダ州ボカ・ラトンで誕生した。IBM は当初パソコン市場をあまり重要視していなかった。しかし、WordStar や VisiCalc の当時によって徐々に企業に導入されるようになりつつあった。しかし、パソコン市場は Apple II と CP/M によってほぼ独占されようとしていた。つまり、IBM は早急に市場に参入する必要があった。そこで、IBM は、自社所有技術の利用という IBM の規定に従うのではなく、手っとり早く市販の標準品を集めてマシンを構築するという戦略をとった。そのため、ハードウェア的には極めて凡庸なマシンができあがった。OS は Microsoft の DOS を採用し、CPU にはインテルの 8088、周辺チップも基本的にはインテルの標準品を採用した。それまで、IBM は最先端技術を創造し続け業界を牽引してきただけに、雑誌等では落胆的な記事が多かったという。

IBM は、コアのハードウェア製品 (インテル製 CPU) あるいはコアのソフトウェア製品 (Microsoft 製 DOS) を統制する排他的契約を締結していなかった。インテルと Microsoft は IBM 互換機を製造しようとする他の企業に販売することができた。また、オープンアーキテクチャを採用することによって仕様が公開され、サードパーティ⁶が周辺機器や互換

⁶この用語は元々は IBM の社内用語

ソフトウェアを開発販売し易くしていた。既製品で構成されていたため、当時としては破格の安値 1500 ドルで販売された。

こうして、IBM PC 互換機ビジネスが登場した。ご存じのように、IBM PC 互換機市場には多くのサードパーティが参入し、瞬く間にパソコン市場のデファクトスタンダードとなった。

CNET の PC 25 周年を振り返るインタビュー (デル (2006)) で、デルコンピュータのマイケル・デルが、「もし IBM が PC に対して厳重なコントロールを保持し続けていたとしたら、PC 業界はどうなっていたと思うか」という質問に対して、

IBM が PC を厳重にコントロールする方針を採用していたら、市場は今よりずっと小さく、進化のスピードもはるかに遅いものになっていただろうということと言えます。

と答えている。

そして、さらにマイケル・デルは、「今日の PC の形成に役立った要因は何か」という質問に対して、半導体技術の飛躍的進歩とエコシステム効果をあげている。

エコシステム効果ともいえる要因があります。つまり、IBM の PC 戦略と Intel、Microsoft の連携によって実現された業界環境です。このエコシステムは、文字通り、数万社におよぶ企業と数十億というユーザーによって形成されています。Dell は世界中に 2 億台を超える PC を販売しており、2006 年度の販売台数も 4 千万台に達しています。ユーザーと関連企業の協調によって形成されたエコシステムは、極めて大きな影響力を持ち、単なる 1 企業ではとても実現できないことを実現できます。

マイケル・デルの意見については、IBM を含めて多くの人が同意するであろう。

この経験により、IBM は、自社単独で市場を開拓するよりも、オープンなプラットフォームを提供してサードパーティを巻き込む方が大きな市場を創出できることを体験したはずである。

ただし、この経験は IBM にとって苦い経験となった。元々自分で始めた互換機市場の主導権をみすみす Intel と Microsoft に握られてしまったからである。この教訓が Eclipse エコシステムにどのようにいかされているかについては次章で解説する。

3.4.2 Linux オープンソース活動への参加

Eclipse をオープンソース化したもうひとつの理由を、IBM の Linux でのオープンソース活動の中に見い出すことができる。

IBM は Eclipse で自らオープンソースプロジェクトを立ち上げたわけであるが、それ以前に既存のオープンソースプロジェクトに参加してオープンソースとは何かを学習してい

る。その経緯が、ウィキノミクス(タブスコット, ウィリアムズ (2007)) に詳しく解説されている。

IBM は 1998 年ごろ、新しい OS の開発をためらっていた。なぜなら、OS 開発には莫大な費用がかかり、しかも市場が受け入れてくれるとは限らないからである。ちょうどそのころ、Linux が勢力を増しており、IBM にとっては魅力的な選択肢に映った。Linux コミュニティに参加する前に、IBM の資金で Apache ソフトウェア財団を設立し、IBM は Apache プロジェクトへの参加を財団と契約した。そこでは、オープンソースの文化、開発プロセス、考え方を予習した。そうして、オープンソースの驚異的なスピードと生産性を知った。

その価値はどの程度のものであったのか。

チェスブロウ (2007) には、IBM の戦略部門のバイスプレジデント、ジョエル・コーリーの発言として IBM が Linux に参画して得た価値が語られている。

長い間、商業的に有効なオペレーティング・システムを作成し、維持していくには 5 億ドルはかかると考えてきた。今日、我々は毎年約 1 億ドルを Linux の開発に費やしている。そのうち約 5000 万ドルは Linux の信頼性向上のための基本的改善に費やされている。残りの 5000 万ドルは IBM が必要とする機能、たとえば特定のハードウェアやソフトウェアと接続するためのドライバー・ソフトなどの開発に使う。我々は OSDL (Open Source Development Lab) に問い合わせ、他社における Linux の商業目的の開発投資額は概算でどのくらいなのか聞いた。大学や個人の作業は除き、我々と同じ企業のみのものである。OSDL によると、投資額は年間約 8 億から 9 億ドルであり、そのうち基本的ニーズへの対応作業と特定ニーズへの対応作業の割合はほぼ五分五分のことだった。有効なオペレーティング・システムを開発するために必要だった 5 億ドルに相当する金額が (特定ニーズ部分を除き基礎ニーズ部分のみを考慮した場合) 今も Linux 開発にかかっている。そこには、我々は 1 億ドルしか払っていない。財務面に限ってみても、これが我々にとって有利なビジネスであることが理解できる。(チェスブロウ (2007) P242)

オープンソースであれば、独自開発していた場合にかかっていたはずの金額の 20% の予算で済む。開発コストを大幅に削減でき、開発が驚異的なスピードになる。

以上のように、IBM は、Apache と Linux ですでにオープンソース活動を経験しておりノウハウがあった。さらには、オープンソースによる投資対効果が非常に大きいことも知っていた。だから、オープンソース化に踏み切ったのであろう。

3.4.3 オープンな市場なら勝てる

大工であれ、プロ野球選手であれ、一般的に、慣れた道具を手放すのはなかなか難しい。これは、ソフトウェアの開発環境にも当てはまる。一端、道具 (ツール) に慣れてし

まうと、なかなか他のツールを試すことができない。他のツールの操作性等を習得するまでの学習コストが少なからずかかるため、一時的にせよ生産性が落ちるからである。

ところが、IBMは相互運用可能なツールのオープンな市場が登場すれば、開発者がツールを取っかえ引っかえすることが容易になるだろう言う。GUIなどの共通する機能についてはプラットフォームが提供してくれているため、ツールが変わったとしても違和感なくすぐに操作できるようになることを指しているのであろう。そのようなオープンな市場はどのベンダーにとっても公平であるが、IBMはその市場で十分に競合できる自信があるという。

OTIマーケティングのリーダーである Marc Erickson はインタビュー Erickson and Brody (2001) の中で次のように述べている。

IBMは、相互運用可能なツールのオープンな市場があれば必ず役に立つはずだと確信しています。こういったアプローチであっても、製品がより有用になり、より開発者のプロジェクトと容易に統合できるようになるため、すべての人が恩恵を受けることとなります。真にオープンな開発環境は、非常に活気のある市場が形成され、そこでは人々はプロプラエタリーなシステムにロックインされることはなく、IBMと競合他社が参加する市場は公平なものになります。

IBMは自社製品が機能では競合できると自負しているので、公平な市場というのはIBMにとって吉報です。企業が一度与えられたツールセットに束縛されてしまっていると、なかなか他のツールには移行しにくいものです。これがオープンな市場であれば、製品は機能本位で競合することができ、それはIBMが望むところです。

これまで開発者はプロプラエタリーなツールに囲い込まれていた。だからIBMは勝てなかった。しかし、開発環境のオープンな市場が登場すれば、IBMは実力本位で勝負でき、勝てる。開発者の周りを囲っていたプロプラエタリーなツールの壁を一端破壊し更地にして、そこで改めて競争をやり直す。そうすれば勝てる。これが、Mark Ericksonの、そしてIBMの主張である。

以上が、IBMの目論見のひとつであろう。

3.5 まとめ

リサーチクエスションは、「IBMが2001年になぜ4000万ドルもの資産価値のある開発ツールを無償のオープンソースとして寄贈したのか」であった。

2001年はちょうどMicrosoftの.NETがエンタープライズ向けアプリケーション市場に参入しようとしていた時期であった。ところが、Java陣営はSunの市場拡大戦略のため、Java陣営内部で競合関係にありまとまっていなかった。特に、.NETと比較すると開発環

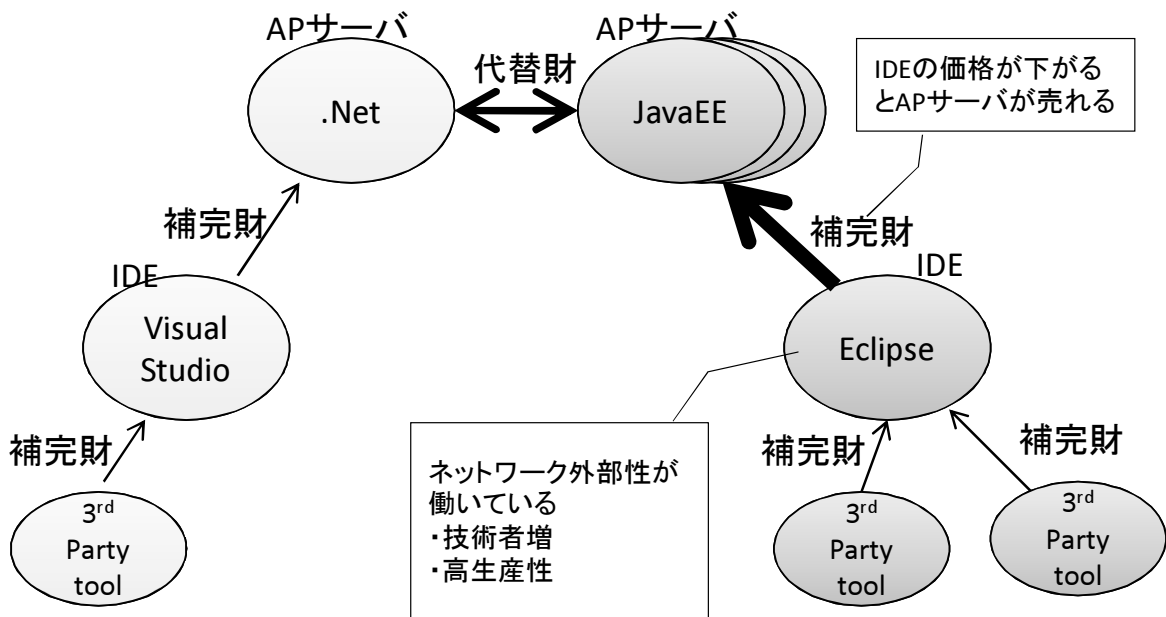


図 3.2: Eclipse のミクログ経済的構造

境に弱点があった。開発ツールがうまく連携できていなかったのである。早急に、Javaの開発環境を強化する必要があった。

そこで、IBMは一計を案じ、開発環境のプラットフォームを無償のオープンソースとして提供することとした。これによって、各社でバラバラであった開発ツールが統合され、補完財であるAPサーバが.NETに対抗できると目論んでいたようだ。

開発環境の無償オープンソース化は、ミドルウェアの強化・ソリューションカンパニー化というIBMの戦略にもマッチしていた。また、IBMにはオープンソースでの経験があり、勝算もあった。

以上が、リサーチクエストに対する解である。図で示すと図 3.2 のようになる。

次章では、Eclipseの発展過程を追いながら Eclipse エコシステムの構造について論ずる。

第4章 Eclipseエコシステムの進化の過程

4.1 概要

前章では、Eclipse エコシステム誕生の背景について論じた。この章と次章で、Eclipse エコシステムの仕組みについて論ずる。

2001年にEclipseエコシステムが世に登場したが、当然のことながら最初から現在の形で出現したわけではない。状況に応じて少しずつ変態を繰り返した結果、現在の形に進化した。この章では、IBM PCエコシステムをベンチマークとして、Eclipseエコシステムの進化の過程を追いながら、その仕組みを明らかにしていく。

研究にあたっては、インタビュー記事、eclipse.orgの情報(技術情報と参加ベンダーの情報)、その他資料・参考文献を調査した。

本章の構成は以下の通りである。

4.2節では、IBM PCエコシステムの成功要因を分析する。

4.3節では、モノとしてのEclipseの仕組みについて、その背景を明らかにする。元々、どういう考えがあって、ソフトウェアとしてのEclipseのアーキテクチャが生まれたのかを探る。

4.5節では、制度としてのEclipseの仕組みについて、2004年の改革の内容を明らかにする。エコシステムの活動を最大化する定期リリースについて、4.6節で説明する。

4.2 IBM PC エコシステム成功要因

この節では、まず、IBM PCが広く普及した理由について、IBM PC互換機のアーキテクチャとパワー、クスマノ(2005)の指摘を解説し、IBM PCエコシステムの成功要因を分析する。

4.2.1 オープンな仕様とプラグインアーキテクチャ

3.4.1節で前述したように、IBMは、コアのハードウェア製品(インテル製CPU)あるいはコアのソフトウェア製品(Microsoft製DOS)を統制する排他的契約を締結していなかった。インテルとMicrosoftはIBM互換機を製造しようとする他の企業に販売することが

できた。また、オープンアーキテクチャを採用することによって仕様が公開され、サードパーティ¹が周辺機器や互換ソフトウェアを開発販売し易くなっていた。すなわち、サードパーティの参入障壁は極めて低かった。

IBMは、最初のIBM PCからIBM PC/AT²まで、IBMがアーキテクチャを設計して公開するオープンアーキテクチャ戦略を採用し続けた。オープンアーキテクチャを採用したことによって、アーキテクチャだけではなく、マザーボードの形状、電源、ケースの形状まで、全くIBM PCと同じクローン機を製造する弱小ベンダーが登場してきた。さらには、小さなパソコンショップが、少ない投資で台湾ベンダーから部品を仕入れて互換機を組み、販売することができるようになった。

次に、ハードウェアのアーキテクチャについて考察する。

IBM PCは、マザーボードがオンボードI/Oをほとんど持たず、I/Oデバイスを拡張カードとして実装する形式を採用していた。フロッピーディスク、パラレルポート、ディスプレイ表示機能(グラフィック・カード)は、拡張カードを用いて実装されていた。このおかげで、ユーザ自ら簡単にサードパーティ製品と交換することができた。また、サードパーティが特別な用途の特殊I/Oデバイスを開発したとしても、IBM PCの拡張スロットにプラグインすることもできた。

以上より、次のことが推論できる。

- プラグインアーキテクチャが、周辺機器市場を創生する
- オープンな仕様が、サードパーティの参入を促進する

周辺機器とはすなわちプラットフォームにとっての補完財である。

この2つによって、数多くの互換機メーカーが参入し、瞬く間に互換機市場が立ち上がることができたのだろうと分析する。

4.2.2 プラットフォームリーダーの交代

1984年にIBM PC/ATを発表したところまでは良かったが、その後、成長が停滞してしまった。理由は、PCアーキテクチャの陳腐化(ガワー, クスマノ(2005))が始まったからである。

インテルのマイクロプロセッサはムーアの法則に従って急速に進化し続けていた。これに対して、PCのアーキテクチャはIBM PC/ATアーキテクチャ以来長らく変化していなかった。特にISAバスの性能が大きなネックになっていた。インテルのマイクロプロセッサが持つ能力が、古いアーキテクチャの影に隠れて、ユーザがその恩恵を享受できていなかったという。インテルがプロセッサの性能を向上させたのと同じ速さではPCのプラッ

¹この用語は元々はIBMの社内用語

²1984年に発表した後継機。AT互換機という言葉のルーツとなった。

トフォーム自体が進化していなかった。ガワー、クスマノ (2005) によれば、インテル幹部には相当不満がたまっていたらしい。

満を持して、IBM は PS/2 を発表するが、バスを ISA とは互換性のない MCA に変更した。そのため、これまでの周辺機器はそのままでは使えなくしてしまった。しかも、MCA の仕様は公開せず、高いロイヤリティを取るクローズド戦略への回帰を図った。その結果、IBM は PC 業界からの反発を買ってしまう。

インテルは PC の一部品メーカーにすぎなかったが、IBM のこの行動に業を煮やし、IBM に代わって、PC 業界に新しい PCI バス・アーキテクチャを提案した。この瞬間に、PC アーキテクチャのプラットフォームリーダーが IBM からインテルに入れ代わった。

ご存じのように、インテルは PC の繁栄を謳歌し、IBM は PC の生みの親であるにもかかわらずパソコン事業から撤退することになった。

以上から得られる示唆は、エコシステムの進化のスピードと同期してプラットフォームも進化すること、互換性を維持すること、公平感を醸成すること、が重要であるということである。公平感とは、エコシステムへの参加者が提供する価値と獲得する価値のバランスが公平と思えることをいう。

進化スピードの同期と互換性の2つは、エコシステムへの参加者が互換性を維持しながら同じスピードで進化していくという意味で、共進化という言葉を使用すれば、エコシステムが継続的に発展していくためには、次の2点が重要であると言える。

- エコシステムの共進化
- 公平感の醸成

4.2.3 エコシステム成功因子

以上をまとめると、IBM PC 互換機の成功には、次の重要な4つのポイントがあり、これらが、IBM PC 互換機エコシステムを急速に立ち上げと継続的な発展に寄与したと推測する。

1. プラグインアーキテクチャ
2. オープンな仕様
3. エコシステムの共進化
4. 公平感の醸成

以降では、Eclipse の進化の過程を追いながら、こらら4つのポイントが Eclipse にはどのように実装されていたのかをみていく。

4.3 1998年：開発ツールとしてのプラットフォーム

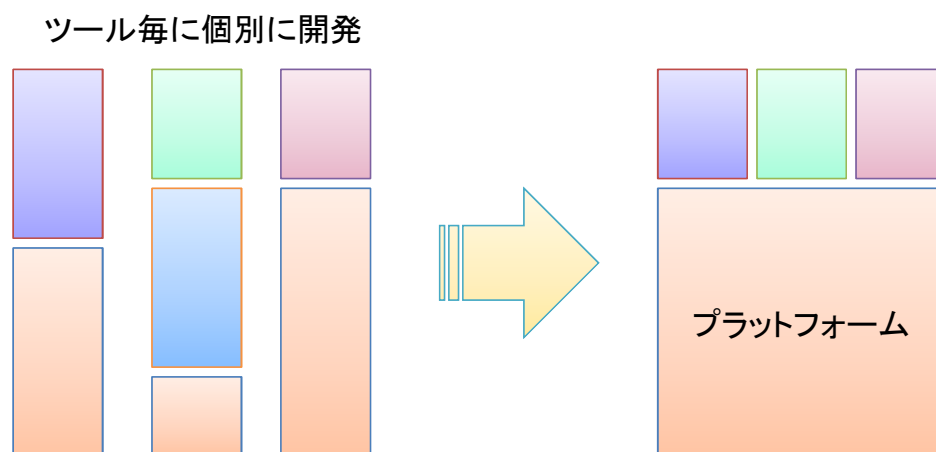


図 4.1: 開発ツールの共通機能のプラットフォーム化

ソフトウェアの開発では、開発言語や開発対象に合わせて開発環境を用意する。開発環境はそれぞれの目的に最適化されているので機能的に少しずつ異なる部分もあるが、共通する部分も多い。たとえば、ほとんど開発環境にはエディタというソースコードを編集する機能があり、また、ソフトウェアのバージョン管理機能、ビルド機能などは大抵のソフトウェア開発には必要な機能である。

ところが、共通する機能があるにもかかわらず、開発言語が異なると開発環境を開発する担当部署も異なり、それぞれ別々に同じような機能を開発するはめに陥ることが多い。また、複数の開発ツールを独立して開発しているために、シームレスに統合されておらず使い勝手が悪くなってしまいうという課題もある。

こうして、IBM 社内では開発ツールの共通機能をプラットフォームとして提供する機運が高まった(図 4.1)。Cernosek (2005)によると、そもそも Eclipse は、IBM の開発ツール製品の二重開発を回避するためには、共通機能を集約したプラットフォームが必要であるという IBM の内部的な理由により 1998 年 11 月からプロジェクトが始まった。

最初のターゲットは Java 用の開発環境 (IDE) だった。IBM が 1996 年に買収した OTI (Object Technology International) 社は IDE 開発について経験が豊富であり、こじんまりとしては高いスキルのチームを持っていた。OTI はコアとなるプラットフォームの開発を担当し、IBM はそのプラットフォーム上に機能を追加した開発ツール製品に取り組んだ。

このプラットフォームが完成すれば、二重開発を回避できるだけでなく、共通のプラットフォーム上に構築されたツールどうしがスムーズに連携することができ、さらには基本的な操作性についてもプラットフォームが提供してくれるため操作性も向上する。

こうして、Eclipse のソフトウェアアーキテクチャ(図 4.2) が次のように設計された。

- 共通機能はプラットフォームとして提供し、追加機能はプラグインとして実装。

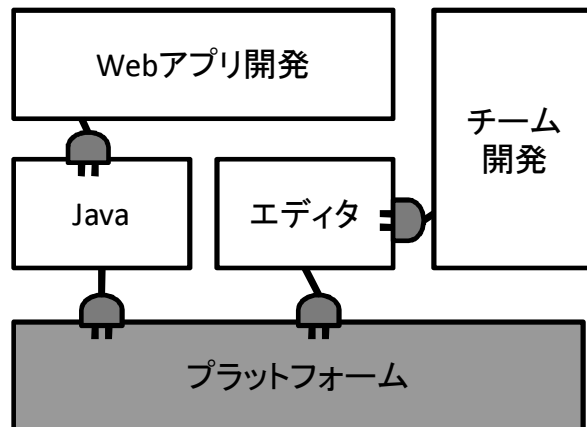


図 4.2: Eclipse プラットフォームのアーキテクチャ

- プラットフォームの機能を利用するための API を定義。
- API を利用してプラグインを開発。プラグイン開発のためのツールも提供。

Eclipse プラットフォームは、ツールをシームレスに統合するための仕組みと、その仕組みを利用するためのルールを API という形式で開示している。拡張機能は、この API にしたがってプラグインという形で実装する。API にしたがって実装されたプラグインを、Eclipse プラットフォームに追加すれば、新たな機能を追加したことになる。追加したプラグインの操作性は統一されており、かつ、プラグインとプラットフォーム、さらにはプラグイン同士がシームレスに連携することもできる。さらに、プラグインを開発するためのフレームワークや開発環境 PDE(Plug-in Development Environment)、ドキュメントも多数提供されている。

ある程度のプログラミング能力が必要ではあるが、PDE をダウンロードすれば誰でもプラグインの開発が行える。しかも、エディタなど、開発環境としての一般的な機能はプラットフォーム側が提供してくれているので、拡張したい機能だけに集中することができる。開発環境は非常に複雑なソフトウェアであるため、あるアイデアを思いついたとしても、開発環境としての基本的な機能の実装に多くの時間が費やされてしまい、なかなか思いついたアイデアを実装するには至らなかった。ところが、Eclipse の登場により、プラグイン機能の開発に集中できるようになった。その結果、大学の研究室やベンチャー企業などが斬新なアイデアをプラグインとして実装できるようになった。

すでにお分かりのように、上記のプラグインアーキテクチャは、IBM PC の拡張カードのプラグインアーキテクチャと同じ設計思想である。

- 1998 年に実現されていたエコシステムの要素
 1. プラグインアーキテクチャ

4.4 2001年: Eclipse 誕生

4.4.1 オープンソース化

前章で解説したように、IBM は Eclipse のオープンソース化を決断し、2001年に公開する。

Cernosek (2005) によれば、「顧客にとって最も理想的な開発環境というのは、IBM だけではなく、顧客のカスタムツールやサードパーティのツールをも組み合わせた環境である」と考え、「ヘテロな環境ではあるが互換性があるツール環境がソフトウェアツールのエコシステムの発端となった」という。

Eclipse プラットフォームは、次の基本要件を満たすような設計思想で作れるようになった (Erickson (2001))。

- アプリケーション開発用のさまざまなツールの構築を支援
- 独立系ソフトウェアベンダーを含めてサポートするツールベンダーは無制限
- 任意のコンテンツ・タイプを操作するためのツールを支援
- 同じコンテンツ・タイプやツール・ベンダーのツールだけでなく、異なるコンテンツ・タイプまたはツール・ベンダーのツールについても、シームレスな統合を容易化
- Windows や Linux を含めて、広範囲の OS で動作
- 開発ツールの作成には広く普及している Java 言語を利用

IBM PC との対比した場合、オープンソース化は、IBM PC のオープンアーキテクチャと同じ思想である。オープンソースとして提供することによって、仕様とソースコードが公開され、しかも誰でも無償で拡張機能 (プラグイン) を開発することができる。

4.4.2 Eclipse コンソーシアム

第1回議事録³によると、Eclipse の最初のミーティングは、2001年11月29日、シカゴのオヘア国際空港のアメリカンエアラインの提督クラブの会議室で開催された。会則やメンバーシップ、組織、議長、世話人⁴等について承認を行った。当初、Eclipse はコンソーシアム形式でスタートした。

創立時のメンバーは、Borland, IBM, Merant, QNX, Rational, Red Hat, TogetherSoft, SuSE, WebGain であった。恐らく、IBM は開発ツールの主要ベンダーに、Eclipse への参加を打診したと思われる。当時、日立へは、日本 IBM から Eclipse への参加の打診があっ

³<http://dev.eclipse.org/viewcvs/index.cgi/www/org/board/minutes/nov2001.pdf?root=Eclipse.Website&view=co>

⁴当初は、Eclipse のボードメンバーの担当者を世話人 (Steward) と呼んでいた。

たそうだ。Eclipse コンソーシアム設立後のメンバー数の推移は図 4.4 の通りである。詳細は付録 B を参照されたい。

議事録や Web 上の資料⁵を調べると、Eclipse に参加する際には、各社、プロポーザルを宣言することになっていたようである。プロポーザルの中では、次の 2 点について明言している。

- Eclipse プラットフォームの上に自社製品を開発すること
- Eclipse に対する貢献

例えば、富士通の場合、富士通の開発環境 APWorks を Eclipse ベースにする予定であること、Eclipse で COBOL プロジェクトを立ち上げ貢献すること、の 2 点が記載されている。

参加ベンダーに価値の獲得と貢献を同時に求めることによって、「共進化」の実現を狙ったのであろう。

プラグインアーキテクチャに加えて、2001 年にはオープンな仕様と共進化を実現していた。

- 2001 年までに実現されていたエコシステムの要素
 1. プラグインアーキテクチャ
 2. オープンな仕様
 3. 共進化

4.5 2004 年: Eclipse の改革

オープンソースのカルチャーをエンジンとして、開発ツールのプラットフォームである Eclipse は、強力なエコシステムを構成するはずであった。ところが思ったほど、メンバー数が伸びなかったようだ。そこで、IBM は制度改革を実施する。

4.5.1 誤算と原因

Eclipse がリリースされてから 2003 年までの間、Eclipse は世界中の Java 開発者に瞬く間に広がった。日本でも同様の現象がみられた。IT 系の Web マガジンである@IT⁶が実施した 2003 年の調査 (小柴 (2003)) では、

⁵例えば富士通のプロポーザル: http://dev.eclipse.org/viewcvs/index.cgi/www/org/eclipse_fujitsu/?root=Eclipse_Webst

⁶<http://www.atmarkit.co.jp/>

Eclipse は 2001 年 11 月にオープンソース化された新しいツールであるだけに、前回 (2002 年 5 月) 調査時点での使用率はわずか 1% にすぎなかった。それがたったの 1 年で、JBuilder など先行する諸製品を追い抜き、“もっとも使われる Java IDE ”の座に就いてしまった。また読者が“今後使いたい”開発ツールを見ても、Eclipse の数字はダントツに高くなっている (図 4.3)。

とレポートしている。その翌年 2004 年の同じ調査ではほぼ 50% が Eclipse を利用するまでに伸びた。Eclipse はまさに「野火」(小柴 (2003)) のように広がった。

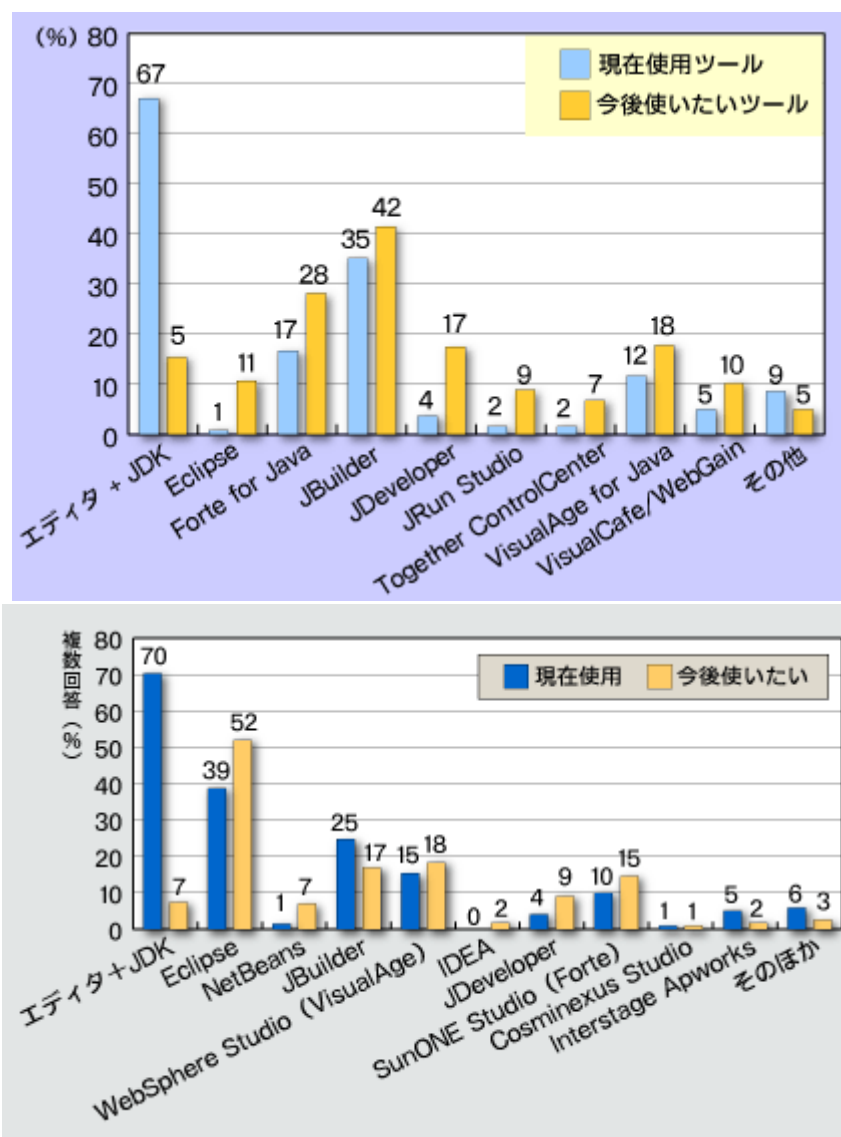


図 4.3: Java 開発ツールの使用状況 2002 年 (上段) と 2003 年 (下段)(小柴 (2003) より引用)

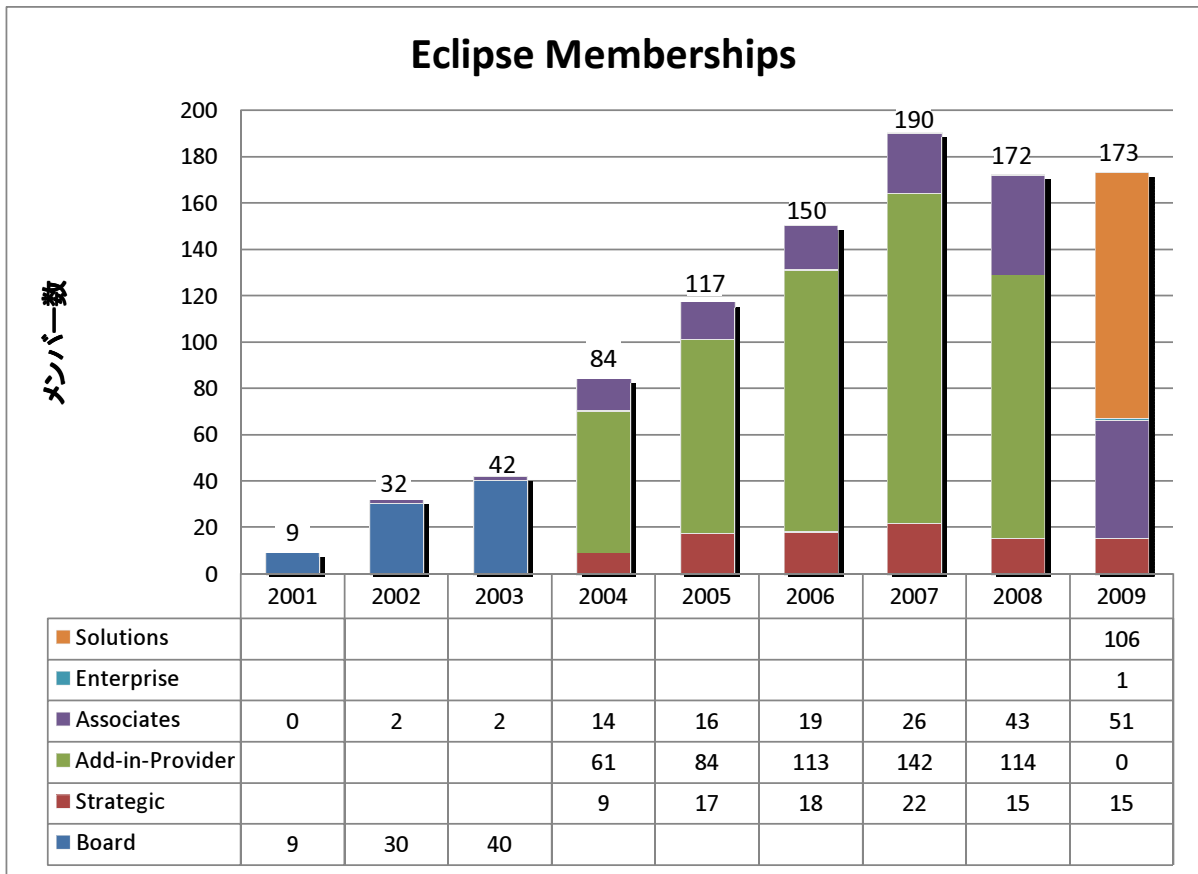


図 4.4: Eclipse のメンバー数の変化 (付録 B より集計)

ところが、Eclipse コンソーシアムに参加するメンバー企業数はそれほど伸びていなかった(図 4.4)。

業界アナリストは、市場が Eclipse は IBM の統制下にあると認識していると分析していた。要するに、「不公平感」を感じていたのである。

この認識によって大手ベンダーが戦略的に Eclipse にコミットするのを躊躇しているという。他のベンダーに真剣にコミットして欲しいのであれば、IBM とは切り離して Eclipse をより独立した組織に再編すべきであった。(Cernosek (2005))

このレポートを受けて、Eclipse は IBM とは独立した NPO として Eclipse 財団に再編成され、EclipseCon 2004 で発表された。

4.5.2 改革内容

EclipseCon 2004 で発表された Eclipse の主な改革は、次の 2 点である。

- Eclipse を IBM から独立した NPO 組織とすること
- ライセンスを CPL(Common Public License)⁷から EPL(Eclipse Public License)⁸に変更すること

NPO 組織化

規制に関しては、NPO 組織とするために会則を大幅に変更した。会則 (Bylaw⁹) の先頭には Eclipse の目的が記載されており、しかも第一文にはベンダー中立であることが真っ先に宣言されている。

Section 1.1 Purposes.

The Eclipse technology is a vendor-neutral, open development platform supplying frameworks and exemplary, extensible tools (the “ Eclipse Platform ”). Eclipse Platform tools are exemplary in that they verify the utility of the Eclipse frameworks, illustrate the appropriate use of those frameworks, and support the development and maintenance of the Eclipse Platform itself; Eclipse Platform tools are extensible in that their functionality is accessible via documented programmatic interfaces. The purpose of Eclipse Foundation Inc., (the “ Eclipse Foundation ”), is to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services. The Eclipse Foundation is formed exclusively as a non-profit trade association, as set out in section 501 (c) (6) of the Internal Revenue Code (the “ Code ”).

すなわち、

Eclipse 技術の目的とは、ベンダー中立かつオープンな開発プラットフォーム (“Eclipse プラットフォーム”) を提供することである。Eclipse プラットフォームとは、フレームワークであり、他のツールの雛型となる拡張可能なツールである。

Eclipse 財団の目的とは、Eclipse プラットフォームの創造、進化、プロモーション、サポートを前進させ、オープンソースコミュニティと補完的製品・機能・サービスのエコシステムを育成することである。また、Eclipse 財団は独立非営利団体とする。

⁷<http://www.eclipse.org/legal/cpl-v10.html>

⁸<http://www.eclipse.org/legal/epl-v10.html>

⁹<http://www.eclipse.org/org/documents/Eclipse BYLAWS 2003.11.10 Final.pdf>

会則の改革によって、Eclipse はベンダー中立なプラットフォーム提供に専念するとし、業界アナリストに指摘された不公平感を払拭して「公平感の醸成」を図ったものと推測する。

ライセンスの変更

Eclipse メンバー (HP 社) からのクレームにより、CPL ライセンスに含まれていた特許報復条項を削除した。それが EPL ライセンスである。

削除されたのは第 7 項の一文:

If Recipient institutes patent litigation against a Contributor with respect to a patent applicable to software (including a counterclaim or a claim in a lawsuit), then any patent licenses granted by that Contributor to such Recipient under this Agreement shall terminate as of the date such litigation is filed.

参考訳¹⁰

受領者がコントリビューターに対し、交差請求や反訴を含めて、ソフトウェアに適用される特許に関して特許訴訟を起こした場合、そのコントリビューターから当該受領者に付与された特許ライセンスは、その訴訟が正式に起こされた時点で終了するものとします。

この一文は、受領者 (すなわち Eclipse の利用者) が、コントリビューター (Eclipse にソースコードを寄贈した者) に対して、特許訴訟を起こした瞬間に、利用料無料の特許ライセンスが停止することを意味する。この文章のポイントは、特許訴訟の対象を「ソフトウェアに適用される特許」としていることであり、ソフトウェアであればどのソフトウェアでも構わないことである。つまり、Eclipse 以外のソフトウェアについて特許訴訟を起こした場合でも、Eclipse の利用を停止させることができる。Eclipse を広く開発現場に適用している場合、まったく Eclipse とは関係のないソフトウェアに対して特許訴訟を起こすと、開発が止まってしまう。これでは、制約が強く、Eclipse のコントリビューターどうしが特許訴訟を起こすと Eclipse の開発自体が停滞してしまうと、HP 社が強く懸念を表したと聞く。

EPL ライセンスへの変更によって、参加ベンダーにとっての訴訟リスクが軽減される。参加ベンダーの不安要素が減り、エコシステムへの参入障壁を下げることになる。

この「安心感の醸成」は、IBM PC エコシステムの 4 つの成功要因にはなかった要素である。

以上のように、2004 年は参入障壁であった不公平感を解消するために、ライセンスと組織を改革した。

¹⁰http://sourceforge.jp/projects/opensource/wiki/licenses%2FCommon_Public_License

- 2004年までに実現されていたエコシステムの要素

1. プラグインアーキテクチャ
2. オープンな仕様
3. 共進化
4. 公平感の醸成
5. 安心感の醸成

4.6 2006年：定期リリース

オープンソースソフトウェアに限らないが、特にオープンソースでは次のバージョンがいつリリースされるのかがはっきりしない。たとえ、リリース日が決まっても期日通りにリリースされないことがある。そうすると、その日に向けてビジネス戦略を立てることが難しくなる。また、Eclipseのようにプラットフォームにプラグインが依存しているようなソフトウェアでは、プラットフォームとプラグインのバージョンが同期していないと、折角、イノベーションによって新機能が提供されたとしても利用することができない。

Eclipseでは2006年より、Eclipseほとんどのプロジェクトが同期して毎年6月末に一斉にバージョンアップ版をリリースするようになった。Eclipseではこれを「リリーストレイン」と呼んでいる。

この結果、Eclipse関連の製品やサービスを扱っているベンダーは計画を立てやすくなった。また、Eclipse財団の発表によれば、プラットフォームと各プラグインが同期して開発を進めるため、バグも発見されやすくまた迅速に収束するようになったという。

IBM PCエコシステムでは、インテルやマイクロソフトのプラットフォーム製品リリースに合わせて、製品を開発しリリースするが、同日ということはまずない。よって、プラットフォームが最新になっても、補完財は古いバージョンであるため、最新機能を十分に享受できないことがあり得る。

ところが、Eclipseでは、エコシステム全体として完全に同期をとったかたちで同日リリースする。そのため、リリース日から、プラットフォームの最新機能を利用したプラグインが使えるというメリットがある。

2006年になって、「共進化の醸成」が強化された。

4.7 まとめ

この章では、IBM PC互換機のエコシステムをベンチマークとして、時間軸に沿って追いながら、Eclipseエコシステム発展の施策を明らかにした。

施策を表 4.1 にまとめる。IBM PC 互換機のエコシステムと比較した場合に、大きく異なるのは、「安心感の醸成」という要素が追加されたことである。しかも、これは Eclipse エコシステムを創始した IBM からではなく、参加ベンダーである HP 社からでた要望であることが、非常に興味深い。つまり、以前の CPL ライセンスは、HP 社にとってはリスクと映っていたということである。

次の章では、Eclipse のメンバーにとって、これらの施策の何が魅力と映っているのかをインタビューを通して探求する。

表 4.1: IBM PC と Eclipse エコシステムの施策

エコシステム成功因子	IBM PC エコシステム	Eclipse エコシステム
プラグインアーキテクチャ	拡張ボード	Eclipse プラグイン
オープンな仕様	オープンアーキテクチャ	オープンソース
共進化	技術の進歩にあわせてプラットフォームを設計	価値の獲得と貢献を宣言
	互換性維持	定期リリース
公平感の醸成	ベンダー中立	会則でのベンダー中立の宣言
安心感の醸成	—	ライセンスの変更(特許報復条項の削除)

第5章 インタビュー

5.1 概要

この章では、前章で整理したエコシステム発展の施策のうち、参加ベンダーにとっては何が魅力に映っているのかをインタビューによって明らかにする。

5.2節で、インタビュー結果をまとめ、その後の各節においてインタビュー結果について推論を展開する。

5.2 インタビュー調査:参加の理由

Eclipse エコシステムの施策が、Eclipse の参加ベンダーにはどのように映っており、そのうちのどれが魅力で Eclipse に加盟したのかを明らかにするために、2009年3月に開催された EclipseCon 2009¹ において、インタビューを試みた(詳細は付録 A を参照)。

対象は、EclipseCon で出会った Eclipse のメンバー企業である。特に、ベンチャー企業を中心にヒアリングした。

インタビューの結果、主たる Eclipse の参加動機として次のような理由を聞くことができた。

- Eclipse が市場として魅力であること
- 開発コストの削減が期待できること
- IP 管理プロセスが整備されていること
- EPL ライセンスがビジネスに向いていること

以降では、上記のインタビュー結果について推論を展開する。

5.3 Eclipse 市場が理由

Eclipse の市場は巨大かつワールドワイドである。例えば、図 5.1 は、Eclipse 財団がレポートしたダウンロード数の推移であるが、最近ではおおよそ毎月 100 万ダウンロード

¹<http://www.eclipsecon.org/2009/>

はある。また、図 5.2 は、ダウンロード数を地域ごとに集計したものである。このグラフより南米は相対的に利用が少ないものの、世界中で Eclipse が利用されていることが分かる。特に、アジア地域での利用が活発である。

翻って、ベンチャー企業の立場になってみれば、Eclipse のインストールベースの数やワールドワイドな広がり是非常に魅力的に映るのは至極当然である。

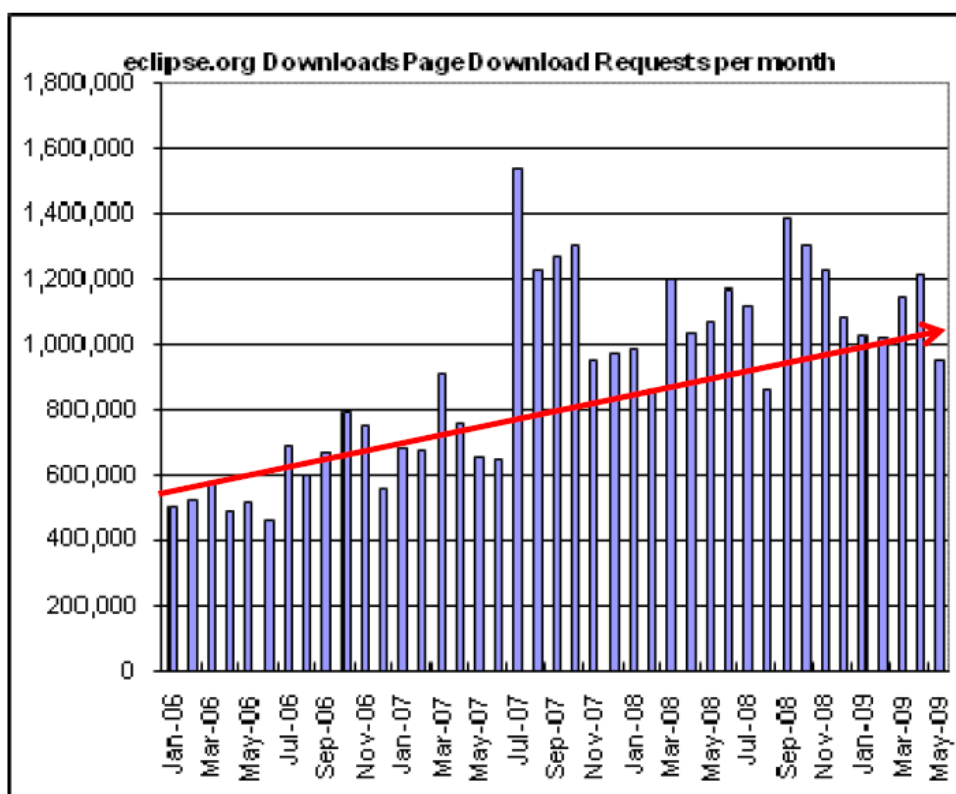


図 5.1: Eclipse ダウンロード数の推移 (Eclipse Foundation (2009) より引用)

Eclipse と同等の Java 用オープンソース開発環境に NetBeans がある。これは、Sun が中心となって推進しているが、Eclipse と比べると利用者数が少ない。ProteCode 社に参加理由をインタビューしたときに、はっきりと

NetBeans と比較して、単純にユーザベースの違いで Eclipse に参加することにした

と語っていた。

SOA(Service Oriented Architecture, サービス指向アーキテクチャ) 関連のドイツのベンチャー企業 Sopera は、Eclipse に参加した理由を次のように語ってくれた。

ユーザコミュニティにアプローチする最も効率的な方法が Eclipse へのコードの寄贈だった。寄贈したソースをベースに Eclipse にコミュニティ(プロジェク

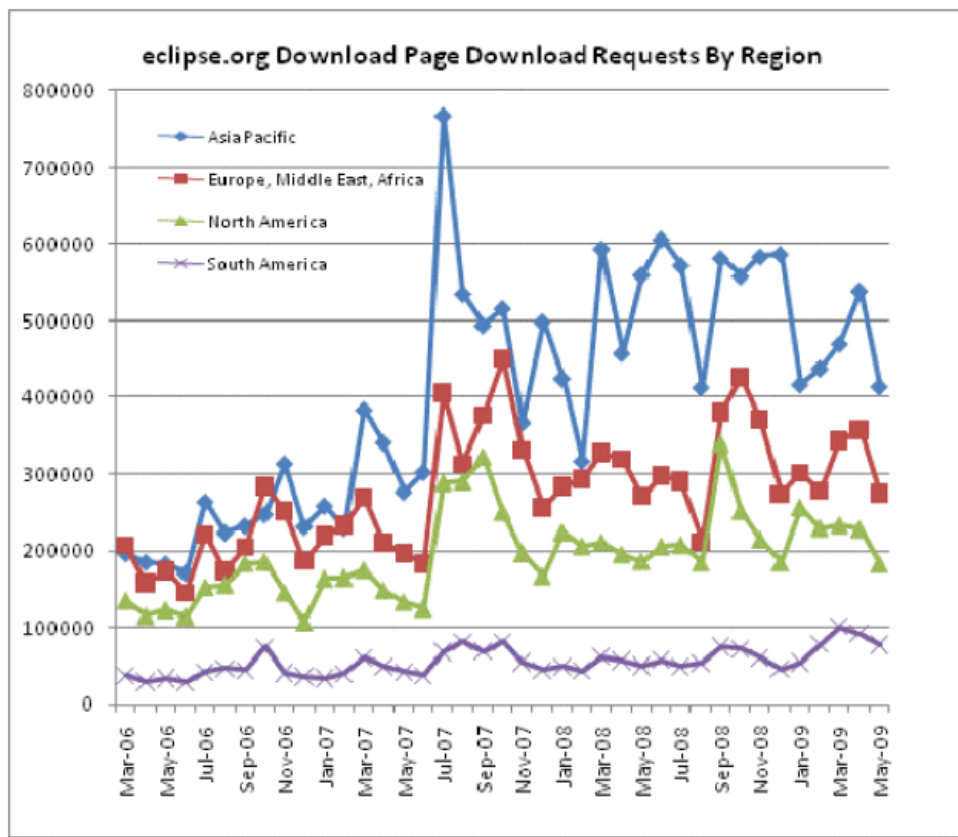


図 5.2: 地域ごとのダウンロード数の推移 (Eclipse Foundation (2009) より引用)

ト)を立ち上げ、コミュニティの進む方向をリードする。ビジネスは、ソフトウェアそのものではなく、その上位のサポートやコンサルティングなどのサービスで行う

SOA の市場には IBM と Oracle という巨大競合が既に存在する状況で、小さな、それも、北米以外のベンチャー企業が参入するのは一般的な方法ではほぼ困難である。Eclipse にソースコードを寄贈することによって、ユーザ市場に比較的容易に浸透することができる。インタビューで直接語った上記の目論見意外に、次のような効果も期待しているであろう。

- Eclipse という高い知名度をもったブランド力を活用できること
知名度が低くても Eclipse のサイトやイベントでの社名とソースコードの露出度が高まる。
- 無償のオープンソースとすることによって直接ユーザにリーチできること
- 競争のレイヤーをプロダクトからサービスにシフトできること

サービスを受ける側にとってみれば、オープンソース版 SOA のソースコードを熟知したベンダーの方が信頼できる。

Eclipse に積極的に参加して、成功した企業に Actuate 社がある。EclipseCon 2009 のセッションの中で Actuate 社が Eclipse の市場開拓力についてレポートしていた。Actuate は BI(Business Intelligence) ツールのベンダーで、ビジネスデータの可視化ツールを開発している。Eclipse への参加にあたっては、可視化のエンジン部分をオープンソースとして寄贈した。その結果、Actuate の知名度が向上するとともに、市場への浸透・新市場の開拓に成功したという。

Actuate によれば、2008 年のダウンロード数は 650 万ダウンロード以上で、そのうちの 1% が有償の顧客となってくれているという。1% という小さいようだが、ダウンロード数が大きいので、それでもビジネスとして成り立つ。また特に印象的だったのは、「Actuate がそれまでリーチできていなかった市場へも参入することができた」という発言である。それまで Actuate は金融市場がメインの市場であり、他の業種にはほとんど進出できていなかったという。ところが、Eclipse にソースコードを寄贈し、コミュニティ(プロジェクト)をリードしたことによって、流通業、製造業、テレコム業などへも参入できるようになった。Eclipse に参加しなければ恐らく参入することはまずなかった市場だという。同様に地理的にもこれまでリーチできていなかった海外市場、例えばインドや中国などのユーザを獲得することもできたという。

このような成功事例があれば、Eclipse が持つ市場のポテンシャルはますます魅力的に映る。

また、ユーザ市場ではなく、Eclipse 技術者の市場に惹かれて教育ベンダーが Eclipse のメンバーになっていることも面白い発見であった。AvantSoft は、インド系の技術教育ベンチャー企業であり、主に技術者を対象として Eclipse や Java に関する教育プログラムを提供している。教育プログラム自体の開発はインドで行っているようであるが、主に北米の技術者を対象としており、e-ラーニングや研修などさまざまな教育プログラムを用意している。

以上、インタビューより、無償のオープンソースが持つ市場開拓力や市場創出力が、参加ベンダーには非常に魅力に映っているという事実が確認できた。

5.4 開発コスト削減が理由

開発コストには 2 種類ある。新たな機能を開発する新規開発コストと、既存機能のバグを修正する保守コストである。前者は、共通機能を他のベンダーと共同開発することによって削減することができる。後者は、できるだけ多くのユーザに利用してもらうことによって、網羅的なテストが可能になり、利用するユーザが多ければ多いほど、多くのバグを検出することができる。保守コストは、バグの修正もさることながら、バグの検出にもかなりのコストがかかるため、コスト削減になる。Webtide 社は、開発者ベースが増え

ることを期待して Eclipse に参加したという。そもそも Webtide は jetty というオープンソースを Apache 2.0 ライセンスで提供しており、ある程度の成功を納めている。それが、今回、Eclipse にも参加し開発プロジェクトを立ち上げ、Apache 2.0 と EPL のデュアルライセンスで提供することとした。実は、Eclipse も jetty もベースに OSGi² という技術を利用しており、Eclipse の周りに OSGi の開発者が集結しつつある。実際、EclipseCon には最新の OSGi 技術情報を求めて参加している人も多い。したがって、Webtide は単なる一般的な意味でのソフトウェア開発者を求めているのではなく、OSGi 技術を持った開発者増を期待しているのであろう。つまり、特定の技術のメッカとして Eclipse があり、新技術や開発者を求めて Eclipse に集結するという構図が見える。

一方、保守コストの削減に期待して参加を決断したベンチャーもある。Sopera もそんな一社である。インタビューにおいて、Sopera の CTO は保守コストの削減と明言していた。「オープンソース化することによって、多くの人が利用してくれるため、多くの不具合が早期に発見される。」どれぐらいの保守コスト削減を期待しているのかとさらに聞いたところ、「約 50% 程度の削減を見込んでいる」という答えであった。

参加ベンダーどうしがうまく協力し合うことができれば、開発コストや保守コストを低減させることができる。インタビューイの口から直接聞くことはできなかったが、実は、Eclipse にはコミュニティ(プロジェクト)を立ち上げるための支援サービスが定められている。オープンソースコミュニティを立ち上げるのは簡単ではない。オープンソースコミュニティ育成についてノウハウを持った Eclipse が支援してくれれば安心できるのではないだろうか。

以降で、コミュニティ(プロジェクト)育成支援サービスについて詳しく解説する。

5.4.1 コミュニティ育成支援サービス

Linux の成功をみて誰もが、オープンソースの市場創出力、イノベーションのパワー、開発生産性の高さ、開発スピードの速さを実感し、うまく活用したいと夢見たはずである。ところが、自らオープンソースプロジェクトを独自に立ち上げてコミュニティを形成するのは、ノウハウも乏しくなかなか難しい。例えば、ネットスケープ社は 1998 年に主力製品であるブラウザのソースコードをオープンソースとして公開したが、コミュニティの立ち上げに失敗した。この教訓について、ネットスケープのオープンソース化の強力な推進者であったジェイミー・ザウンスキーは次のように述べている。

何にでも効くような万能薬ではないということだ。もし、ここに教訓があるとすれば、死にかけていたプロジェクトに「オープンソース」の魔法の粉を振りかければ、すべてが魔法のようにうまくいくなどということはないということだ。(ウェバー (2007) P165)

²<http://www.osgi.org/>

日本では、ニュートーキョーが、1999年に「セルベッサ」と呼ぶ業務アプリケーションをオープンソース化するという野心的な試みがあったが、ソースコードを公開しただけで体制が整えられておらず活性化しなかった(湯澤(2005) 竹田, 米山(2002))。

以上のように、ソースコードを公開さえすればコミュニティが立ち上がるというものではない。そこで、Eclipseでは、Eclipse傘下でプロジェクトを立ち上げるための支援の仕組みが整備されている(表 5.1)。以下に、仕組みを列挙する。

- プロジェクトのインキュベーションプロセスが明文化されている。
これによって、傘下企業の責任範囲が明確化される。
- 組織体制が確立している。
責任と役割が明確になっている。インキュベーションプロジェクトの責任はテクノロジートッププロジェクトのPMC(Project Management Committee)が持つ。
- リスク軽減策が講じられている
プロジェクトの設立時にはメンターが割り当てられ、プロジェクト育成のアドバイスを受けることができる。また、インキュベーションプロセスの途中のチェックポイントでレビューを実施することになっている。
- オープン性・透明性の確保
プロジェクト設立時には、プロポーザルを公開し、プロジェクトの趣旨、スコープ、ロードマップ等を明らかにし、参加者を募る。また、レビュー結果や議事録も Web上で公開する。
- IT インフラの支援
プロジェクトの Web サイト、ソースコードのリポジトリ、課題/バグ管理など、プロジェクト運営に必要な IT インフラは Eclipse 財団が用意してくれている。

Eclipseでは、以上のような手厚いサービスを提供することによって、参加ベンダーの安心感を醸成し、積極的にコミュニティ(プロジェクト)を立ち上げてもらいたいと意図していると推測する。

コミュニティ(プロジェクト)を立ち上げるときのプロポーザルの中で、必ず価値を共有することを宣言しなければならない。

Eclipseではプロジェクトを始める場合に、プロポーザルを掲示してプロジェクトの賛同者を募ることになっている。プロポーザルでは、必ずプロジェクトの目的を明記するが、Eclipseの会則 Section 1.1 に従わなければならない。すなわち、「ベンダー中立かつオープンな開発用プラットフォーム」を提供しなければならない。プロジェクトの成果物は、ある目的に特化したツールであってはならず、特定のベンダーに依存しない拡張可能なプラットフォームでなければならない。つまり、成果物は、他者も活用できる共有の財産とすることを宣言しなければならないのである。

表 5.1: Eclipse のインキュベーションプロセス (Duenas et al. (2007) からの引用)

	Eclipse
インキュベータ プロセス	<ul style="list-style-type: none"> ・ 2003/11 にプロセス制定 ・ 法的文書に明記 ・ 原則が公開されている
組織構造	<ul style="list-style-type: none"> ・ 役割と責任が明確に定義されている ・ 中央集権型の構造: テクノロジートッププロジェクトの PMC がインキュベーションプロジェクトの責任を持つ
リスク軽減方法	<ul style="list-style-type: none"> ・ プロジェクト設立レビュープロセスの初期段階では、EMO(Eclipse Management Organization) が、コミュニティの規模を査定する。プロポーザルフェーズでは、トップレベルプロジェクトとするかサブプロジェクトとするかを定める ・ 最初の安定リリースまでのインキュベーション/バリデーションでは対話的アプローチをとる
進行状況の公開	<ul style="list-style-type: none"> ・ プロポーザルと作成レビューはまとめられて、Eclipse プロポーザル Web サイトで公開 ・ チェックポイントレビューはまとめられて、Web 上に公開 ・ 定期的なプロジェクトレビューと議事録は Web 上で公開
プロジェクト 支援のインフラ	<ul style="list-style-type: none"> ・ プロジェクト Web サイト ・ コードリポジトリ ・ ダウンロードサイトとミラーサイト ・ メール管理環境 ・ イシュー/バグ・トラッキング

5.5 IP 管理プロセスが理由

参加理由の回答で以外だったのは、Eclipse の IP 管理プロセスがしっかりしていることを理由に上げるベンダーが思っていたよりも多かったことである。Meristic 社も理由にあげていたし、Webtide 社は真っ先に IP 管理プロセスが整備されていることを理由に上げていた。

オープンソースを利用する場合に不安になるのが、ライセンス違反と特許である。

オープンソースライセンスの中でも GPL ライセンスが混入しているとやっかいである。なぜなら、「GPL のライブラリー」を「GPL ではないプログラム」と動的リンクした場合、「GPL ではないプログラム」も GPL にしなければならないからである。GPL ライセンスのソースが混入するとすべてのソースコードが GPL に連鎖的に汚染されてしまう可能性がある。実際、プロプライエタリな製品の一部に、GPL ライセンスのオープンソースを使用していたために、最終的に製品のすべてのソースコードを開示することになってしまったベンダーが何社も存在する。

他社特許の混入もやっかいな問題である。製品のベースとしたオープンソースに他社の特許がいつのまにか混入していたために訴えられるというケースがある。Linux では SCO 社が、自社の特許が Linux に使われているとして、Linux のユーザである DaimlerChrysler と米 AutoZone を提訴したことがあった (ITpro (2004a))。最近では、プログラム中のデータとデータベース中のデータとのマッピングを行うオープンソースの ORM(Object Re-lation Mapping) ソフトウェア Hibernate に自社の特許が使用されているとして、Oracle、RedHat, HP, Dell, Genuitec が訴えられている (InfoWorld (2009))。オープンソースソフトウェアは自社でスクラッチから開発したわけではないので、どこの誰とも分からない開発者が素性の知れぬソースコードをコピー&ペーストしている可能性を排除することができない。その上、通常、訴えられるのは、混入した開発者ではなく、特許が混入したソフトウェアを利用・販売しているベンダーである。さらに、特許訴訟は時間と費用がかかるものであるので、ひと度、特許訴訟が起これば、特にベンチャー企業にとっては経営上のインパクトが大きい。ITpro (2004b) によれば、300 万ドルかかるという。

オープンソースでビジネスをするベンダーには、常にこのような不安があるはずである。ところが、Eclipse の場合は IP 管理プロセスが整備されており、汚染されないような仕掛けが施されている。ベンダーの不安を取り除き、安心感を与えてくれる。

Eclipse 財団もこの点を理解しており、New Member Jumpstart セッション³の中で、Eclipse 財団が推測するメンバーになる理由として、

- メンバーのうち 1/3 – 1/2 は、それまで数万ドルかかっていた開発環境の費用が無料になり、浮いた分を寄付したいため
- IP 管理プロセスが整備されているから

³Eclipse のプロモーションのためのセッションであり、参加を検討しているベンダーを対象に Eclipse のメリットを解説する。

- Eclipse ブランド (ロゴ) を利用できるから
- EclipseCon などのイベントでプレゼンスを向上できるから
- よい人材を雇用しやすいから
- 他のメンバーへの情報を配信してもらえるから

を挙げており、IP 管理プロセスを独立した項目として列挙している。つまり、Eclipse 財団は IP 管理プロセスが参加メンバーを惹きつけていることに自覚的である。

5.5.1 IP 管理プロセス:クリアランス手続き

Eclipse では前述した課題に対処するために、知財のクリアランスを制度化している。この制度のおかげで、Eclipse が提供するオープンソースには、GPL ライセンスのソースコードや第三者の特許が混入することはなく、かつ、入っている特許は無償許諾が保証される。

以下では、Eclipse において知財のクリアランスを次の 2 つによって実現しているのかをみていく。これら 2 つは、いわば法と法の運用に当たる。

- EPL ライセンス (法)
- 知財管理プロセス (運用)

EPL ライセンスによって、特許の無償許諾が保証される。

EPL ライセンス⁴の第 2 項 b 条には次のように明記されている。「コントリビュータ」とは「プログラム」を頒布する個人または法人のことを指し、「受領者」とは「プログラム」を受け取るすべての者を指す。

本契約書 (注:EPL ライセンス) の条項に従って、各コントリビューターは受領者に対し、ライセンスされる特許に基づいて、ソースコード形式であれオブジェクトコード形式であれ、当該コントリビューターのコントリビューションを作成したり、使用したり、販売したり、販売用に提供したり、インポートしたり、その他の方法で移転したりする、非独占的で世界規模で使用料無料の特許ライセンスを付与します。

上記の条項は、コントリビューターは受領者に対して無料の特許ライセンスを与えることを意味する。したがって、EPL ライセンスのソフトウェアは常に特許ライセンスを徴収されることがない。

Eclipse では、ソースコードはリポジトリで管理される。リポジトリとはソースを保管・管理するシステムである。リポジトリにソースコードを登録する作業をコミットと呼ぶ

⁴<http://www.eclipse.org/legal/epl-v10.html>

が、Eclipse ではコミットする前に “Eclipse Leagal Process”⁵ と呼ばれる審査手続きを経なければならない。この手続きによって、EPL 以外のライセンスや訴訟リスクのあるコードが混入するのを回避している。

この手続きでは、次のようにソースコードを審査していく。

1. 100%スクラッチから開発したコードの場合は、混入の余地がないので EPL ライセンスでコミットする。
2. EPL ライセンスのソースコードをベースとして開発した場合も、混入の余地がないのでそのままコミットする。
3. 上記以外の場合は、EPL ライセンスで提供可能かどうかを EMO(Eclipse Management Organization、Eclipse 管理組織) が判断する。その場合、IP(知財) 追跡システムに登録し、どのような過程を経て、どう判断したのかの記録(ログ)を取る。
4. もし、EPL ライセンスを設定できないと判断された場合は、当該ソースコードを削除する。

以上の手続きによって、知財関連のクリアランスが実施される。これによって、安心して、訴訟リスクのないソースコードを利用できるようになる。

IP 管理プロセスを設計した Janet Compbell にもインタビューをおこなった。「開発者に負荷がかからないように配慮している」という彼女の発言が特に印象的であった。

5.6 EPL ライセンスが理由

オープンソースにおいて、ライセンスとビジネスモデルは非常に密接な関係にある (West and Gallagher (2006))。したがって、適切なオープンソースライセンスを選択しないと、ビジネス自体が成立しなくなってしまうこともあり得る。例えば、先述した GPL ライセンスでは可能なビジネスの範囲が限られてしまう。

以上のような理由から、Meristic 社は参加の理由として、真っ先に EPL ライセンスがビジネス向きであることをあげていた。

以降では、EPL ライセンスがなぜ魅力的に映るのかを詳しく解説する。EPL ライセンスには、ビジネスとの親和性の良さに他に、価値共有を推進するという側面もある。

⁵<http://www.eclipse.org/legal/EclipseLegalProcessPoster.pdf>

5.6.1 EPL ライセンスと価値共有

ここでは、Eclipse で採用されているオープンソースライセンス、EPL ライセンスについて解説し、EPL がビジネスとの親和性が良いこと、Eclipse における知識もしくは価値の共有の仕掛けとして機能していることを明らかにする。

ライセンス、開発モデル、ビジネスモデル

EPL ライセンスについて解説する前に、オープンソースライセンスと、開発モデル、ビジネスの関係について整理する。表 5.2 に示すように、ソフトウェアは物理的な実態を持たない純粋な知的生産物 (知財) のひとつである。ソフトウェアライセンスは、利用者に一般に特定の条件下でのソフトウェアの使用を許諾する。見方を変えれば、ソフトウェアライセンスとは知財をコントロールする方法といえる。オープンソースソフトウェアとはソースコードが開示されたソフトウェアであるので、オープンソースソフトウェアでは知的財産が公開されている。オープンソース活動とは、公共の知財の開発活動である。一般的に、共同開発の形態をとる。オープンソースビジネスとは、公共の知財をベースとして行うビジネスをいう。

表 5.2: 知的生産物としてのソフトウェアとライセンス

ソフトウェア	知的生産物 (知的財産) のひとつ
ライセンス	知財をコントロールする方法
オープンソースソフトウェア	公開された知的生産物 (公共の知財)
オープンソース活動	知識 (知財) の共同開発活動
オープンソースビジネス	公共の知財でのビジネス

ソフトウェアライセンスは、オープンソースの開発モデルとビジネスに大きく影響する。すなわち、

- クローズすぎるライセンスは、開発コミュニティが育たない
- オープンが強制されるライセンスは、ビジネス機会が限定される

このため、バランスの取れたライセンスが非常に重要になる。

EPL ライセンス

では、Eclipse のライセンス EPL ではどのようにバランスをとっているのだろうか。数あるオープンソースライセンスの中でも、EPL ライセンスは、IBM の弁護士グルー

プが作成しただけに、内容が非常にシンプルである一方で網羅性も高くよく設計されているという (Rosen (2004))。

ソフトウェアの形式には、ソースコードとオブジェクトがある。ソースコード形式でソフトウェアを開発し、コンパイルしてオブジェクト形式に変換する。オブジェクトはコンピュータが読むための形式であり、人が読むことは困難である。ソースコードはテキスト形式であり、人が読むことができる。ソフトウェアがソースコード形式で提供されるということは、その知的財産が公開されることに等しい。

一般的なオープンソースライセンスでは、ソフトウェアの受領者に許諾される権利と受領者が果たすべき義務が規定されている。Eclipse の EPL ライセンス⁶の規定は次のようになっている。

受領者に許諾される権利 複製権、派生物の作成権、実行権、頒布権、サブライセンス権の、非独占的で世界規模で使用料無料の著作権ライセンス、および、使用料無料の特許ライセンス

受領者の義務 ソースコード形式での提供、EPL と同等のライセンスでの頒布

EPL では、ソフトウェアに関する多くの権利が与えられており、実行、改変、頒布、販売などが可能となっている。さらに、特許があったとしてもライセンス料は無料である。つまり、ソフトウェアを頒布もしくは拡張・発展させる方向への権利が十分に与えられている。

一方、義務は、ソースコードの提供と EPL ライセンスでの再頒布の 2 点である。EPL ライセンスでの頒布しなければいけないということは、EPL ライセンスによって得た権利を頒布先の受領者にも与えなければならないことを意味する。例えば、EPL ライセンスのソフトウェアに改変を加えて新たに派生物を作成した場合、改変したソフトウェアの受領者には著作権ライセンス・特許ライセンスを無償許諾しなければならない。ただし、これは EPL ライセンスのソフトウェアを改変した場合であって、Eclipse プラグインのようにアドオンするソフトウェアについては EPL ライセンスは適用されず、商用ライセンスを始めとして自由なライセンスを設定できる。

Eclipse においては、Eclipse をプラットフォームとしてその上位にプラグインを構築することによって、Eclipse が提供する価値を獲得し、プラグインという形で各自固有の価値を付加することができる。プラットフォームに対して新たな価値、例えば改良などは、ライセンスによりソースコードを公開することが義務付けられているため、結局は全員で価値を共有することになる。すなわち、Eclipse ではライセンスによって価値の獲得と共有の境界が明確に規定されており、この境界に従ってバランスさせることになる。

⁶<http://www.eclipse.org/legal/epl-v10.html>

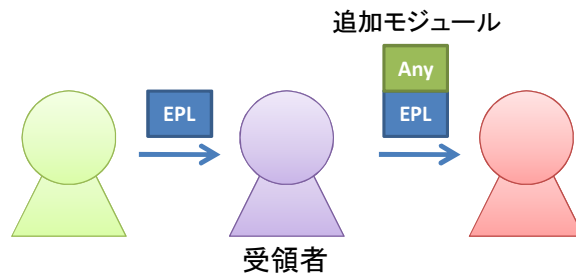


図 5.3: EPL ライセンスの権利と義務

5.7 まとめ

インタビューの結果、参加ベンダーがみている Eclipse 像が明らかになってきた。では、インタビューで明らかになった参加の動機、「Eclipse 市場」「開発コスト削減」「IP 管理プロセス」「EPL ライセンス」とは何であるのか、抽象度を上げて考えてみることにする。

「Eclipse 市場」とは、そこでビジネスを展開しようと考えているということである。Eclipse から利潤を得ようとしている、すなわち、Eclipse から「価値獲得」を行おうとしていることを意味する。そして、「EPL ライセンス」はビジネスとの親和性が良い。

一方、「開発コスト削減」は、Eclipse への参加者と協力して、共通部分のコストは共有しようと考えているということである。つまり、Eclipse において「価値共有」を行おうとしていることを意味する。

Eclipse が「IP 管理プロセス」によって提供するものは、「安心感」である。

以上を表 5.3 にまとめる。

表 5.3: ベンダーが Eclipse エコシステムに参加する理由

抽象概念	インタビュー結果
価値獲得	Eclipse 市場・EPL ライセンス
価値共有	開発コスト削減・EPL ライセンス
安心感	IP 管理プロセス・EPL ライセンス

上記の価値共有と価値獲得は、投資に対して得られるバリュー、つまり投資対効果を意味する。一方、安心感はリスクである。インタビューの結果、分かったことは、ベンダーにとってはビジネスを始める場合の一般的な思考方法、投資対効果がどの程度あり、リスクとして何があるのか、と寸分変わらないということである。オープンソースと言えば、これまではボランティアによる活動というイメージがあったが、Eclipse においては、ビジネスとしてドライに捉えていることが分かった。

次章では、これまでで明らかになった Eclipse エコシステムの仕組みを元に、オープンソース・エコシステムの協創プロセスのモデル化を行う。

第6章 オープンソースエコシステムにおける協創の構造(モデルの提示)

6.1 概要

オープンソースエコシステムにおける協創の構造を明らかにする。

本章では、次のように論を展開する。

6.2節では、Eclipse エコシステムのアーキテクチャを技術と財団の観点で整理する。6.3節では、オープンソースエコシステムの協創のプロセスモデルを提示し、これまでに明らかになった Eclipse エコシステムの仕組みと、インタビューによって明らかになった参加ベンダーの視点から解説する。

6.2 Eclipse のアーキテクチャ

4章で紹介した Eclipse の会則に宣言された Eclipse の目的を整理する。Eclipse の目的は、Eclipse 技術の目的と Eclipse 財団の目的とに大別される。

Eclipse 技術の目的 ベンダー中立かつオープンな開発プラットフォームを提供すること

Eclipse 財団の目的 プラットフォームの創造、進化、プロモーション¹、サポートと、エコシステムの育成

この目的を実現するために、これまで紹介してきたようなさまざまな施策が施されている。図 6.1 に Eclipse のアーキテクチャを技術と財団の視点から整理する。そこには、次の3点を実現するための仕組みが施されている。

- 共通機能についてはプラットフォームとして価値を共有
- 技術以外のことについては Eclipse 財団が支援
- ベンダーは付加価値創出に注力

インタビューによって明らかになったように、仕組みが醸成する価値獲得と共有、安心感に惹かれて、ベンダーが参加している。

¹本稿では詳細に説明しなかったが、メンバー企業が Eclipse をベースとした製品をリリースすると Eclipse の会長の名前でエンドースしてくれるなどの支援策がある。

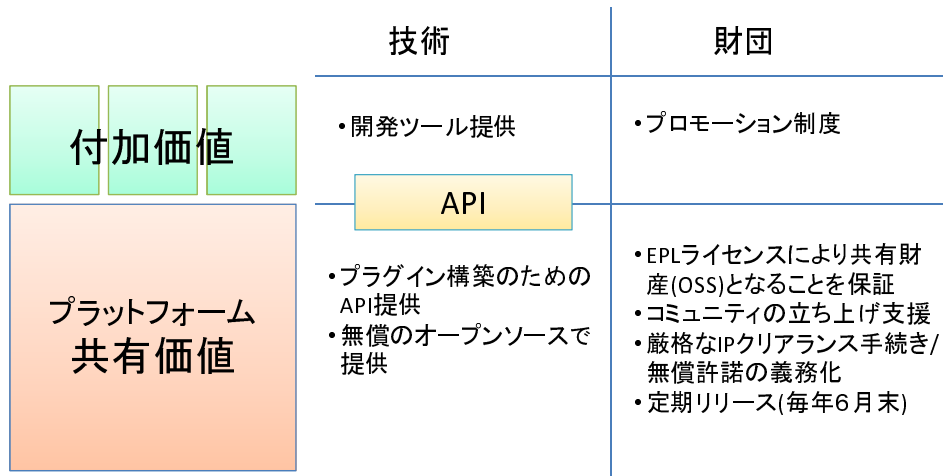


図 6.1: 技術と財団からみる Eclipse のアーキテクチャ

6.3 協創のプロセス

図 6.2 に、オープンソースエコシステムにおける協創のプロセスモデルを図示する。モデル化にあたっては、価値の共有と獲得、質的な発展と量的な発展という軸を導入した。

- 横軸
 - － 価値の共有
 - － 価値の獲得
- 縦軸
 - － 質的な発展
 - － 量的な発展

「量的な発展」とは、新たなベンダーやユーザが参加することによってエコシステムの規模が増すことをいう。これに対して、「質的な発展」とは、イノベーションによって新たな技術や知識といった価値が増していくことを指す。エコシステムは、多様なベンダーが参入し、他の参加者との関係を構築しながら成長していくネットワークである。ネットワーク的観点から説明すると、「量的な発展」とはノード数の増加すなわち参加者の増加を意味し、「質的な発展」とは、ノード間の関係の構築または関係の強化を意味する。

図 6.2 を見て分かるように、上段の「質的な発展プロセス」と下段の「量的な発展プロセス」とがオープンソースプラットフォーム中心として互いに鏡面構造をなす。このモデルで重要なことは、オープンソースエコシステムには、上段と下段の2つのサイクルを廻す仕組みが組み込まれており、サイクルが廻るたびにエコシステムの価値の再生産が繰

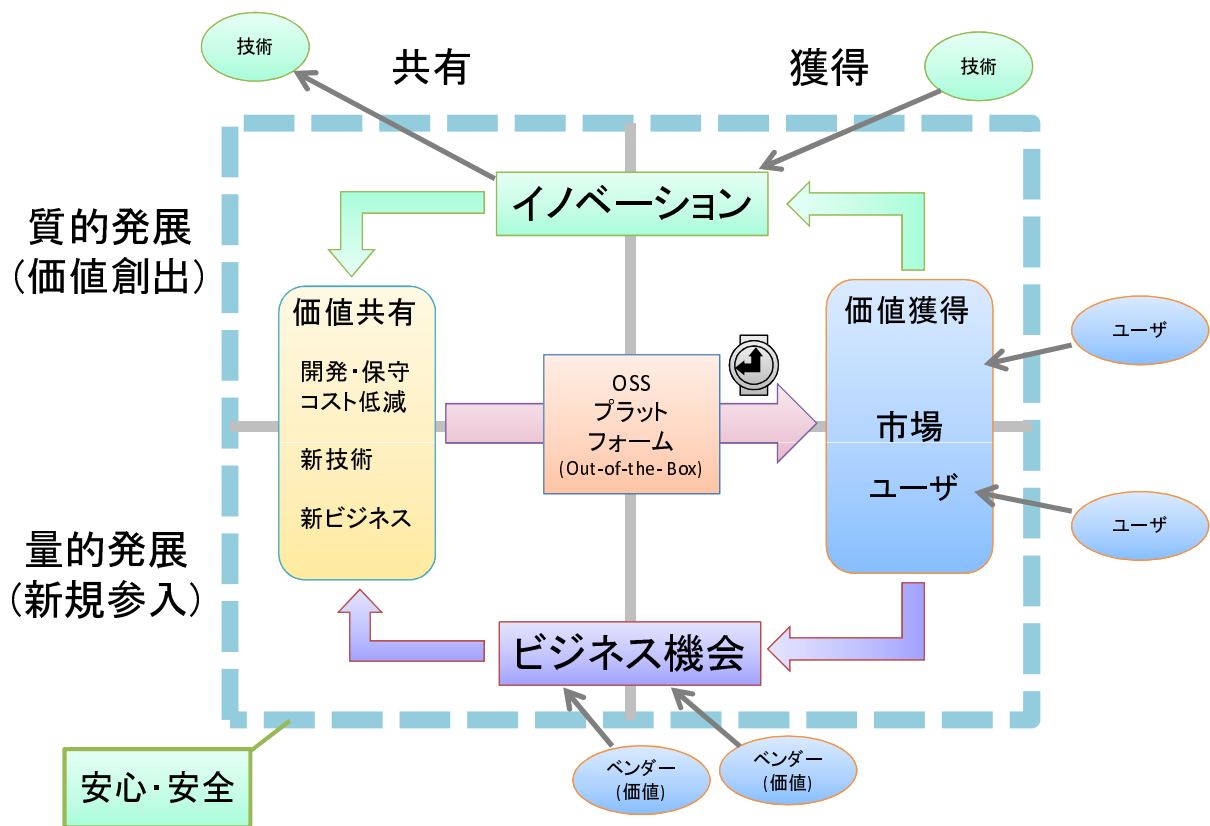


図 6.2: オープンソースエコシステム協創プロセス

り返され、価値が高まって行くことである。これらのサイクルはエコシステム中のコミュニティ(プロジェクト)毎に存在する。それを同期させるのが、4.6節の定期リリースである。毎年、固定日に同時リリースをすることによって、サイクルの回転が最適化される。

以降では、「量的な発展プロセス」と「質的な発展プロセス」、それぞれのプロセスの詳細を解説し、Eclipse エコシステムでは、エコシステムを活性化させるために、Eclipse の何がどのように機能しているのかを説明する。

6.4 量的な発展プロセス

図 6.2 の下側のサイクルが量的な発展プロセスである。参加者、特にベンダーを増やすプロセスである。ビジネス創出のプロセスともいえる。このプロセスでは、優れたオープンソースを求めて、ユーザが増え、増えたユーザが市場を形成しビジネス機会を生む、ビジネス機会を捉えようと新たなベンダーが参加する、という順で推移する。

詳細を説明する。

1. 先ず始めに、Out-of-the-Box な (革新的な) オープンソースソフトウェアが提供されなければならない。
Eclipse の場合、Java 用の開発環境がこれに当たる。Eclipse は、インストールも簡単で、使い勝手がよく、利用すればほぼ間違いなく開発生産性が向上する、非常に優れたソフトウェアである。
2. Out-of-the-Box な (革新的な) ソフトウェアが提供されると、それを使用するユーザが登場する。
ユーザ数を増やすためには、ソフトウェアが優れていることに加えて、ユーザが試用しやすいようにすることも重要である。Eclipse は、無償のオープンソースとして提供され、しかも、世界中のどこからでもダウンロード可能としている。このため、ユーザが Eclipse に参加する障壁が極めて低い。優れたソフトであればユーザがユーザを呼ぶ。
3. ユーザが増えれば、市場が形成され、ビジネス機会が生まれる。
このビジネス機会を捉えようと、ベンダー (ニッチベンチャーなど) が、新たな価値 (機能やサービス) を持ってエコシステムに参加する。ベンダーの中には、ソースコードを Eclipse に寄贈してプロジェクト立ち上げ、市場への浸透と開発コストの削減を狙うものも登場する。その場合に、Eclipse 財団が提供するプロジェクト立ち上げ支援サービスと IP 管理プロセスが参加に対する不安を取り除き安心感を与えていることは、前章で述べた通りである。
4. 価値を共有する。
複数の参加者が協調し合いながら価値を共有する。オープンソースエコシステムの場合、共有された技術的な価値は、ソフトウェアという形で具現化され、ソースコード付きの公共財となる。こうして新たに付加された価値が、新たなユーザを呼ぶ。

以上が、オープンソースエコシステムにおける量的な発展のプロセスである。

6.5 質的な発展プロセス

図 6.2 の上段のサイクルが質的な発展プロセスである。新たな技術や知識を増幅させるプロセスである。知識創造のプロセスともいえる。このプロセスでは、優れたオープンソースを求めて、ユーザが増え市場を形成し、その市場を対象にビジネスを行う、ビジネスから利益が生まれ、新たなイノベーションへの再投資が行われる。

プロセスを仔細に追う。ユーザ増えるところまでは量的な発展プロセスと同じである。

1. Out-of-the-Box な (革新的な) オープンソースソフトウェアが提供される。
2. それを利用するユーザが登場する。

3. ユーザが増えれば市場が形成され、ビジネスを行うベンダーが登場する。
オープンソースを利用するユーザを対象に、付加機能を追加した有償製品を提供したり、コンサルティングや教育などのサービスを提供して利益を上げるベンダーが登場してくる。
4. イノベーションが新たな価値を創生する。
市場から利益を上げることができたベンダーは、さらなる売上増求め、新たなイノベーションに向けて投資をする。または、オープンソースによって技術が公開されると、外部からベンダーや大学・研究所が登場して、その技術に彼らが持つ新たな技術を導入し、イノベーションを起こすケースもある。こうして創出されたイノベーションのうち、プラットフォームはエコシステムに還元され、プラットフォーム以外の部分については独自の付加価値とすることができる。

Eclipse エコシステムに参加するベンダーのイノベーションプロセスは、一般的な企業活動のイノベーションと大きな違いはない。Eclipse をベースとした製品やサービスを市場に提供することによって、売り上げが増え、そこから新たなイノベーションを創生するための投資が行われる。新たに創生されたイノベーションは、次世代の製品やサービスに活かされる。

ここまでは、通常のイノベーションプロセスと同じであるが、最大の違いは、プラットフォームに対するイノベーションはエコシステムに還元しなくてはならないことである。ソフトウェアに対するイノベーションは最終的にはソースコードという形で具現化される。プラットフォームはEPLライセンスで提供されているため、プラットフォームに対するイノベーション(修正)について、EPLライセンスしたがつたソースコード公開の義務が生じる。こうして必ずエコシステムに還元されることになる。いわば、税金のようなものである。

この税金は他者にとっても有益であれば、他者の価値の再生産にも寄与することになる。プラットフォームに関するイノベーションが必ずエコシステムに還元されるという巧妙な仕組みによって、質的な発展のサイクルが生じ、エコシステムは常に発展する構造になっている。

6.6 まとめ

この章では、Eclipse エコシステムをケースとして、オープンソースエコシステムにおける協創のプロセスを、価値の共有と獲得、質的な発展と量的な発展という軸に分けて、モデル化した。

Eclipse エコシステムには、協創のサイクルを生む仕組み、効率よく廻すための仕組み、継続的な成長を維持するための仕組みがうまく機能していることを再確認した。

第7章 オープンソース・エコシステムの課題

この章では、オープンソースエコシステムについて、その構造から導出される課題についてまとめる。

7.1 オープンソースであることによる協創の限界

本稿では、Eclipse エコシステムがオープンソースのパワーをうまく活用してエコシステムを構築してきたことを示した。しかしながら、実は、オープンソースであるが故の限界も存在する。本節では、オープンソースの開発方式を称賛した「伽藍とバザール(レイモンド (1999))」という著名な文書の中に、皮肉にもオープンソースの限界が記されていることを明らかにし、オープンソースエコシステムの課題を論ずる。

伽藍とバザールはエリック・レイモンドがオープンソース開発で観察されていた2つの開発方法論について書かれた文書である。ひとつは伽藍方式 (Cathedral) と呼び、もうひとつをバザール方式 (Bazaar) と呼んだ。伽藍方式とは、企業などで一般的に行われている開発方法で、ある特定の設計者が計画と体制をすべて決め開発するスタイルである。一方、バザール方式とは、中央集権的な管理はなく、あたかもバザーで知らないものどうしが売買をするように、アイデアやソースコードを持ち寄ってソフトウェアを開発するスタイルである。

エリック・レイモンドは、自身のオープンソースプロジェクト `fetchmail` でバザール方式を採用し、成功した。この文書では、そのときに得られた教訓を解説している。

その教訓の中に、オープンソースプロジェクト自体が内包する限界が存在する。

教訓 1. よいソフトはすべて、開発者の個人的な悩み解決から始まる

彼が `fetchmail` の開発を始めたのは、自分が欲しいソフトが世の中に存在しなかったからだ。だから、開発するスキルがあった彼は、自分で開発しようと思立った訳である。

この教訓は、課題を抱えている者とそれを解決する者が同じ開発者である場合にのみ、よいソフトができる、と言ってしまっている。世の中の課題というのは、一般的にそれを

抱えている人と解決してくれる人は別である。したがって、上の教訓が成立するとすると、オープンソースが解決する課題というのは開発者が遭遇する課題に限定されてしまう。確かに、これまで、オープンソースとして大成しているのは、OS、データベース、エディタやメールなどのツールというように開発者が日頃利用している基盤ソフトウェアかそのツールが多い(表 7.1)。これは上記の教訓を裏付けている。

表 7.1: ビジネス化されているオープンソース (West and Gallagher (2006) から引用)

プロジェクト	創立年	創立者	製品カテゴリ	商品化タイプ
Sendmail	1983	UC Berkeley	メールルータ	補完財の販売
Linux	1991	L. Torvalds	OS	共同研究開発
Apache	1995	Eight webmasters	ウェブサーバ	共同研究開発
MySQL	1995	M. Widenius/D. Axmark	DB	補完財の販売
Jike	1998	IBM	Java コンパイラ	スピナウト
Mozilla	1998	Netscape	ウェブブラウザ	スピナウト、 共同研究開発
Darwin	1999	Apple	OS	補完財の販売
Konqueror	2000	KDE プロジェクト	ウェブブラウザ	補完財の販売
OpenOffice	2000	Sun	オフィスソフト	補完財の販売
Eclipse	2001	IBM	開発環境	スピナウト

CIOの記事(CIO (2008))によると、IBMのオープンソース推進者であるボブ・スーター (Bob Sutor) 氏が LinuxWorld Expo & Conference 2008 の基調演説の中で、業務アプリケーションのオープンソースが登場しないことに苛立ちを見せ、

会場の皆さんは、いずれすべてのソフトウェアがフリー・ソフトウェアもしくはオープンソースになる日が来ると信じているかもしれない。だが、それは明日ではないし、おそらく来年でもないし、おそらく10年後でもない。

と話したという。特定産業の業務に詳しく、かつ、その産業専用の業務アプリケーションを開発できるだけのスキルを持った開発者は少ない。「教訓1」が示すように、オープンソース活動の構造上、業務アプリケーションのオープンソースがなかなか登場しないのは、当然の帰結である。個人的な活動をベースとしたオープンソースは、開発者の興味範囲にその成果が限定されてしまうので、すべてのソフトウェアがオープンソースになることは難しい。

また、エリック・レイモンドは、ユーザを持つということは2つの点ですばらしいことだとも指摘している。

一つ目は、

自分が何かニーズに対応しているんだな、なにか役に立つことをしたんだな

とうことを実証してくれることだという。ボランティアとしてオープンソース活動を行うためには、他者への贈与とそれに対する応答がモチベーションを高める。

二つ目が、さらにすばらしいと言っている。

きちんと育てれば、ユーザは共同開発者になってくれるんだ。

(略) ユーザの中にもハッカーがたくさんいるわけだ。そしてソースコードが公開されてるから、かれらは同じハッカーでも役に立つハッカーになってくれる。これはデバッグ時間短縮にはすごく役に立つ。ちょっと励ますだけで、ユーザが問題を診断し、直し方を提案してくれて、一人でやるよりずっとはやくコードを改善できるようにしてくれる。

彼は、この経験から次の教訓を導き出している。

教訓 6: ユーザを共同開発者として扱うのは、コードの高速改良と効率よいデバッグのいちばん楽ちんな方法

だが、これも教訓 1 と同じ理由で、開発者が興味を持つソフトウェアに限定される。ユーザ = 開発者であるような特殊なソフトウェアの場合はこの教訓が成り立つかもしれないが、一般的なソフトウェアではユーザと開発者は異なる。よって、ユーザがどんなにたくさんついたとしても、開発者が集まってこないソフトウェアであればオープンソースの正のサイクルが生じずプロジェクトは萎んでいくことになる。

以上のオープンソースが内包する課題は、Eclipse エコシステムにおいても当てはまる。業務アプリケーションになるほど、ボランティアの開発者は減っていくであろう。これは、Eclipse エコシステムにおいて、ボランティア開発者の偏りが必ず生じることを意味する。偏りが生じ、オープンソースのパワーを期待できない部分において、どうエコシステムを活性化させていくかが課題となる。

7.2 Eclipse エコシステムの末端における協創の限界

この節では、Eclipse の構造上、Eclipse の協創には自ずと限界があるという仮説を示す。

5.6.1 節で説明したように、Eclipse ではプラットフォーム部分は EPL ライセンスによって公共財となることが保証される。ベンダーは、プラットフォームにアドオンする部分を商用ライセンスで販売しビジネスを展開する。Eclipse のプロジェクトを立ち上げて、アドオンするプラグイン部分も EPL ライセンスで提供し、協創を促すケースもある。いずれにせよ、EPL ライセンスによって図 7.1 のような木構造が形成される。したがって、上位が多様化・活性化し技術的なイノベーションが起これば起こるほど、下位のプラットフォームへのソースコードにイノベーションの一部が還元される仕組みになっている。

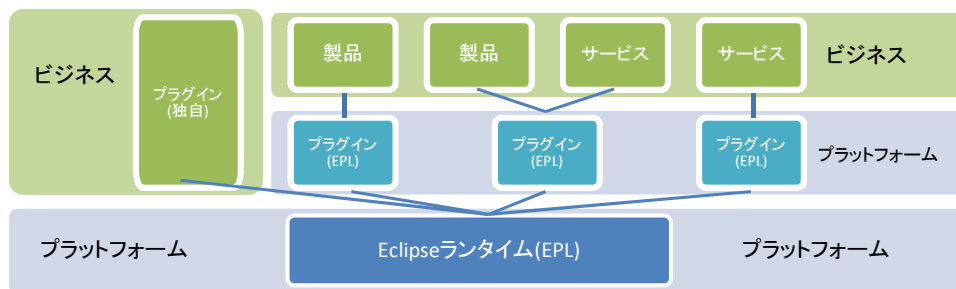


図 7.1: EPL ライセンスによる木構造

ところが、上位のアプリケーションになればなるほど、機能が特化し、興味を持つユーザやベンダーの数は減る。その結果、市場性が小さくなり、ベンダー数も小さくなり、ソースコードのコントリビューションも少なくなるという、悪循環に入り込んでしまう可能性が高い。

仮説: Eclipse エコシステムにおいては、共通プラットフォームから上位のアプリケーションよりになればなるほど、協創が起こりにくくなる。

7.3 量的な発展プロセス不全

6章で解説したように、オープンソースエコシステムでは優れたソフトウェア（プラットフォーム）がオープンソースとして提供されることによって、ユーザが増える。ユーザが自由にダウンロードして無償で利用できるからである。ソフトウェアが優れていれば、ユーザ数が爆発的に増える。そうすると市場として成立するようになり、オープンソースプラットフォームに対して新たな価値（製品機能やサービス）を提供するというビジネス機会が生まれる。このビジネス機会を求めて、エコシステムに新規にベンダー等が参入するというプロセスであった。

ところがユーザの囲い込みを行うベンダーが登場している。Actuate社は、ビジネスデータを可視化するツールのBIRTプロジェクトを立ち上げた。その結果、ユーザの関心を呼び、市場を形成することに成功した。Actuate社は、マニュアルやチュートリアル、フォーラムなど、ユーザがBIRTを学習するための情報を集めたコミュニティサイトBIRT Exchange¹を自社サイト上に構築してしまった。これは、一種のユーザ囲い込みになる。そのため、他のベンダーがオープンソースのBIRTを利用して、ビジネスを始めようとしても、ユーザが囲い込まれているため、なかなか参入に踏み切れない。Eclipseでは、ソースコードについての規定はあるが、ユーザに対する規定はない。

このようにユーザが囲い込まれてしまうと、協創のサイクルがうまくまわらなくなり、継続的な発展が難しくなる可能性がある。

¹<http://www.birt-exchange.org/>

Eclipse 財団は、IT インフラを提供してくれているが、専ら参加ベンダーのためのものである。Eclipse エコシステムの継続的な発展には、ユーザのための IT インフラが必要だといえる。

7.4 まとめ

この章では、Eclipse エコシステムにおける構造上の課題をいくつかあげた。Eclipse はオープンソースを起爆剤として、急激に立ち上げることに成功したが、その一方でオープンソースであるが故の構造上の問題に陥ってしまう可能性もある。

エコシステムが、健全性を維持し続けるには、状況に応じてエコシステム自身が改善するためのメタな仕組みが必要であろう。

第8章 結論

8.1 本研究によって得られた新たな知見

以下に、各リサーチクエスチョンに対する回答を示す。

SRQ1: IBM は Eclipse をなぜ初めたのか

3章で、いくつかの複雑な外的要因と内的要因とがあり、最終的にオープンソースとして提供することがIBMの戦略上合理的な決断であったことを明らかにした。

- 外的要因

- 2001年当時は、Microsoftが.NETを引っさげて、Java陣営の独占であったエンタープライズ向けアプリケーションサーバー市場に進攻しつつあり、脅威であった。.NETとJavaを比較した場合、特に開発ツールが劣っていた。
- Microsoftに対抗する手段としてのオープンソース化は、ガースナーのソリューションカンパニー化戦略に合っていた。

- 内的要因

- IBM社内では、ちょうど開発ツールの二重開発を回避するために、ツールのプラットフォーム開発を1998年から始めていた。
- オープンソース化によってサードパーティを巻き込むことは、ガースナーの顧客指向の発想に適合していた。
- オープンソース文化に対する経験があった。

SRQ2: どのようにEclipseのエコシステムを構築したのか

このリサーチクエスチョンについては、4章で次を明らかにした。

元々、IBM社内の開発ツール製品の共通基盤として始まったEclipseは、次の順序でエコシステムの構築、テコ入れ、最大化を実施してきた。

- プラグインアーキテクチャ

- オープンな仕様
- 共進化
- 公平感の醸成
- 安心感の醸成

SRQ3: Eclipse エコシステムに参加するのはなぜか

5章のインタビューによって、参加ベンダーはEclipse エコシステムの「価値獲得」「価値共有」「安心感」に惹かれることが分かった。これらはとりもなおさず、ビジネスを始める場合に検討する投資対効果とリスクに他ならない。オープンソースと言えば、これまではボランティアによる活動というイメージがあったが、Eclipse においては、ビジネスとしてドライに捉えていることが分かった。

MRQ: オープンソースエコシステムにおける協創の構造は何か

本研究では、Eclipse エコシステムをケーススタディとして考察した結果、オープンソースエコシステムには

- 価値共有と価値獲得
- 質的发展と量的发展

という二つの軸があることが分かった。この軸を使って、オープンソースエコシステムの継続的な発展を促すサイクル、協創のプロセスをモデル化した(図 6.2)。そして、Eclipse エコシステムでは、この協創のプロセスを効率よく廻すための仕掛けが随所に施されていることを明らかにした。

8.2 理論的含意

本研究による理論的含意は、次の2点である。

- オープンソースエコシステムの発展の仕掛けを明らかにしたこと
- オープンソースエコシステムの協創プロセスのモデル化を行ったこと

8.3 実務的含意

本研究の実務的含意は、企業がエコシステムにかかわる場合に次の2つのケースに対して示唆を与えたことである。

- オープンソースエコシステム (Eclipse エコシステム) に参加する場合
共有する価値と獲得する価値を意識しながら、どこでどのように参加ればよいのかという示唆
- 自らエコシステムを構築する場合
どのようにエコシステムを発展させていけばよいのかという示唆

8.4 今後の研究課題

今後の課題のひとつに、本研究が提案したモデルの検証がある。

- 他のオープンソースエコシステムにも適合するモデルであるのかいなか
- Wikipedia のようなオープンナレッジに対して適用できるかどうか
- より一般的なビジネスエコシステムに適用させモデルを洗練する

もうひとつは、定量的な研究が必要だと考える。

今回は、費用対効果を秤にかけているという定性的な事実は分かったが、参加ベンダーがどの程度の費用対効果が見込めたときに参加を決意するのか、定量的なことが全く判明していない。もし、定量的な指標が判明すれば、エコシステムの拡大成長を細かく制御することが可能になるかもしれない。

定量的な分析として、ネットワーク分析もある。エコシステムには、多様な参加者が複雑に絡み合ったネットワークを構成しているため、ネットワーク分析の知見を活かした分析をすることによって、一見しただけでは見えて来なかった事実が判明する可能性がある。

参考文献

- Adomavicius, G., Bockstedt, J., Gupta, A., and Kauffman, R. J. (2006) “Understanding Patterns of Technology Evolution: An Ecosystem Perspective,” in *System Sciences, 2006. HICSS’06. Proceedings of the 39th Annual Hawaii International Conference on*, Vol. 8.
- Cernosek, Gary (2005) “A brief history of Eclipse,”
<http://www.ibm.com/developerworks/rational/library/nov05/cernosek/>.
- CIO (2008) 「【海外 IT 動向】「業務オープンソース・アプリの普及の遅さにいらだつ」
米 IBM の OSS 事業幹部 (2008/08/08) - CIO Online」,
<http://www.ciojp.com/contents/?id=00004789%3Bt=44> .
- Driver, Mark (2002) “Java vs. .NET: Competition or Cooperation?” Technical report, Gartner, Gartner.
- Duenas, Juan C., G., Hugo A. Parada, Cuadrado, Felix, Santillan, Manuel, and Ruiz, Jose L. (2007) “Apache and Eclipse: Comparing Open Source Project Incubators,” *Software, IEEE*, Vol. 24, No. 6, pp. 90–98.
- Eclipse Foundation (2009) “Eclipse Members’ Meeting.”
- Erickson, Marc (2001) “Working the Eclipse Platform,”
<http://www.ibm.com/developerworks/java/library/os-plat/>.
- Erickson, Marc and Brody, Steve (2001) “Interview: The Eclipse code donation,”
<http://www.ibm.com/developerworks/library/l-erick.html>, November.
- Iansiti, M. and Levien, R. (2004) “Strategy as ecology.,” *Harv Bus Rev*, Vol. 82, No. 3, pp. 68–78.
- Iansiti, M. and Richards, G. L. (2006) “The information technology ecosystem: Structure, health, and performance,” *Antitrust Bulletin*, Vol. 51, p. 1.
- InfoWorld (2009) “Red Hat facing JBoss Hibernate-related patent suit | Open Source - InfoWorld,” <http://www.infoworld.com/d/open-source/red-hat-facing-jboss-hibernate-related-patent-suit-250>, March.

- ITpro (2004a) 「米 SCO がついに Linux ユーザーを提訴，自動車業界大手の Daimler-Chrysler と米 AutoZone が標的に：ITpro」，
<http://itpro.nikkeibp.co.jp/free/ITPro/USNEWS/20040304/140880/?ST=oss> .
- ITpro (2004b) 「Linux を利用する企業が特許侵害訴訟を起こされる可能性あり」，米 OSRM が調査結果を発表：ITpro」，
<http://itpro.nikkeibp.co.jp/free/ITPro/USNEWS/20040803/148084/>，8月 .
- Junnarkar, Sandeep (2001) “IBM makes \$40 million open-source offer - CNET News,”
http://news.cnet.com/IBM-makes-40-million-open-source-offer/2100-1001_3-275388.html, November.
- Moore, James F. (1993) “Predators and Prey: A New Ecology of Competition,” *HARVARD BUSINESS REVIEW*, Vol. 71, pp. 75–75.
- Moore, James F. (1997) *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems*: Collins.
- Moore, James F. (2005) “Business Ecosystems and the View From the Firm,” *ANTITRUST BULLETIN*, Vol. 51, No. 1, p. 31.
- Peltoniemi, M. and Researcher, M. S. (2004) “Cluster, Value Network and Business Ecosystem: Knowledge and Innovation Approach,” 『Organisations, Innovation and Complexity: New Perspectives on the Knowledge Economy』 conference, September .
- Riehle, D. (2007) “The Economic Motivation of Open Source Software: Stakeholder Perspectives,” *COMPUTER*, pp. 25–32.
- Rosen, Lawrence (2004) *Open Source Licensing: Software Freedom and Intellectual Property Law*: Prentice Hall PTR.
- Sholler, Daniel (2002) “ZDNet: Web Services / .Net seen gaining steam in dev projects - ZDNet Tech Update,”
<http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2860227,00.html>.
- Vawter, C. and Roman, E. (2001) “J2EE vs. Microsoft .NET,” Technical report, The Middleware Company, pp. 1–37.
- West, J. and Gallagher, S. (2006) “Challenges of open innovation: the paradox of firm investment in open-source software,” *R&D Management*, Vol. 36, No. 3, pp. 319–331.
- アナベル ガワー，マイケル A. クスマノ (2005) 『プラットフォーム・リーダーシップイノベーションを導く新しい経営戦略』，有斐閣，378 頁 .

- エリック レイモンド (1999) 『伽藍とバザール – オープンソース・ソフト Linux マニフェスト』, 光芒社 .
- クレイトン クリステンセン (2001) 『イノベーションのジレンマ 技術革新が巨大企業を滅ぼすとき』, 翔泳社, 第増補改訂版版, 327 頁 .
- スティーブン ウェバー (2007) 『オープンソースの成功 政治学者が分析するコミュニティの可能性』, 毎日コミュニケーションズ .
- ドン タブスコット, アンソニー D. ウィリアムズ (2007) 『ウィキノミクスマスコラボレーションによる開発・生産の世紀へ』, 日経 BP 社 .
- ヘンリー チェスプロウ (2004) 『OPEN INNOVATION - ハーバード流イノベーション戦略のすべて』, 産業能率大学出版部, 209 頁 .
- ヘンリー チェスプロウ (2007) 『オープンビジネスモデル知財競争時代のイノベーション』, 翔泳社, 328 頁 .
- ポール クルーグマン, ロビン ウェルス (2007) 『クルーグマンミクロ経済学』, 東洋経済新報社, 703 頁 .
- マイケル E. ポーター (1999) 『競争戦略論 I/II』, ダイヤモンド社 .
- マイケル デル (2006) 「パソコン 25 年の歴史を振り返る – デル氏に聞く : インタビュー – CNET Japan」,
<http://japan.cnet.com/interview/biz/story/0,2000055955,20202168,00.htm> .
- マルコ イアンシティ, ロイ レビン (2007) 『キーストーン戦略イノベーションを持続させるビジネス・エコシステム』, 翔泳社, 320 頁 .
- ルイス V. ガースナー (2002) 『巨象も踊る』, 日本経済新聞社 .
- 佐々木 裕一, 北山 聡, 国領 二郎 (2000) 『Linux はいかにしてビジネスになったか コミュニティ・アライアンス戦略』, NTT 出版 .
- 小柴 豊 (2003) 「@ I T : Java Solution 第 8 回読者調査結果」,
<http://www.atmarkit.co.jp/fjava/survey/survey0306/java0303.html> , 6 月 .
- 竹田 陽子, 米山 茂美 (2002) 「ビジネス・ケースセルベッサ-ニユートーキョーの食材発注システムはなぜ公開されたのか」, 『一橋ビジネスレビュー』, 第 50 巻, 第 3 号, 146-165 頁 .
- 湯澤 一比古 (2005) 『オープンソースじゃなきゃ駄目』, アイデア出版局 .

付録A インタビュー

A.1 インタビュー概要

2009年3月23日から26日までの間、米国サンタクララで開催されたEclipseCon 2009¹に参加し、インタビューを実施した。EclipseConはEclipse関連では最大のカンファレンスであり、毎年、2月もしくは3月に開催されている。EclipseConには、Eclipse関連事業を行っているベンチャー企業から大手企業、研究者や開発者、コンサルタントなど多種多様な業種・職種の人たちが一同に介し、Eclipseの最新情報について意見交換を行う。

インタビューは、EclipseConの会場で出会ったEclipseのメンバー企業を対象として実施した。

インタビューの目的は、Eclipseへの参加動機を質問することによって、Eclipseの何が参加者を惹きつけるのかを明らかにすることである。

A.2 インタビュー結果

インタビューしたベンダーごとに、以下にまとめる。

- インタビューイの職位
- ビジネス
提供しているEclipse関連の製品もしくはサービス
- 動機/きっかけ
Eclipseのメンバーになった理由

A.2.1 Genuitec

インタビューイ CTO

ビジネス 米国のベンチャー企業。Eclipseと他のオープンソースを組み合わせるパッケージとして提供。ビジネスモデルは、年間30-50ドルのサブスクリプション方式。

¹<http://www.eclipsecon.org/2009/>

動機/きっかけ 2003年に、Eclipseのマーケティング担当者から、Eclipseの新しいプロジェクトWebToolsプロジェクトの設立に招待された。元々、EclipseベースのWebアプリケーション開発用のパッケージを提供していたことが招待された理由。2009年からはStrategicメンバー²になった。理由は、モバイル向けの開発環境のパッケージ化を始めようとしており、ボードミーティングに出席してEclipseのモバイルプロジェクトをリードしたいから。

A.2.2 Innovent Solutions

インタビューイ Vice President

ビジネス 米国のビジネスインテリジェンス等の専門コンサルティング。

動機/きっかけ Eclipseのビジネスインテリジェンスのレポートツール³のプロジェクトBIRTに参加。理由は、BIRTの恩恵を受けていることに対して恩義と責任を感じているため。

A.2.3 AvantSoft

インタビューイ President

ビジネス インド系ベンチャー企業。技術者を対象としたEclipseやJavaに関する教育プログラムを提供している。e-ラーニングや研修など提供形式は多様。教育プログラムの開発はインドで行っている模様。

動機/きっかけ 4年前にIBMからメンバーに参加しないかと打診されたため。戦略的な理由は特に内容であったが、メリットとしてはイベントでの展示やメンバー企業への紹介などの機会が与えられることがある。

A.2.4 Sopera

インタビューイ CTO

ビジネス 2007年に創立したばかりのドイツのSOA⁴関連ベンチャー。SOAPERAというSOA製品販売を中心にサポートやコンサルティングを行っている。

²最上位のメンバーシップ。Eclipseのボードミーティングに参加できる。

³ビジネスデータの可視化ツール

⁴Service Oriented Architectureの略。企業内部のシステムをサービスとして切り出し連携させることを目指したソフトウェアアーキテクチャ

動機/きっかけ 自社製品の核のソースコードを Eclipse の SOA プラットフォームプロジェクト Swordfish⁵ に寄贈し、自身でプロジェクトをリード。Eclipse に参加した理由は次の2つ

1. 売上増

ユーザコミュニティにアプローチする最も効率的な方法が Eclipse へのコードの寄贈、プロジェクトのリード。サービス(サポートやコンサルティング)でビジネスをする。

2. 保守コスト低減

オープンソース化することによって多くの人々が利用するため、多くの不具合が早期に発見される。50%程度の保守コスト低減を期待。

以下は、筆者の推測である。

SOA の市場には IBM と Oracle という巨大競合が既にいる状況で、弱小新参ベンチャー企業が参入するのは一般的な方法ではほぼ困難である。Eclipse にオープンソースとして寄贈することによって、上記のインタビューイが直接語った目論見意外に、次の3つの効果が期待できる。

- Eclipse という高い知名度をもったブランド力を活用できること
知名度が低くても Eclipse のサイトやイベントでの社名とソースコードの露出度が高まる。
- 無償のオープンソースとすることによって直接ユーザにリーチできること
- 無償提供されるので、競争のレイヤーをプロダクトからサービスにシフトできること
サービスを受ける側にとってみれば、オープンソース版 SOA のソースコードを熟知したベンダーの方が信頼できる。

A.2.5 Webtide

インタビューイ CEO

ビジネス jetty という Web アプリケーションサーバを提供する米国のベンチャー。jetty をオープンソースとして提供し、サポートサービス、カスタム化等でビジネス。

動機/きっかけ jetty は Eclipse にソースコードを寄贈するよりも前に、Apache 2.0 ライセンスのオープンソースとして提供し、既にオープンソースとしてのある程度確立した地位にあるのに、改めて Eclipse にソースコードを寄贈して参加した理由は次の3点

⁵<http://www.eclipse.org/swordfish/>

1. IP 管理プロセスが整備されていること
2. 開発者ベースを増やせること
3. すでに Eclipse 内部で jetty を利用していること
Eclipse は内部に Web アプリケーションサーバを持っており、それが jetty である。他のオープンソースプロジェクトであるよりも、Eclipse のプロジェクトになっている方が、不具合報告などのフィードバックを得やすい。

ライセンスは、Apache2.0 と EPL のデュアルライセンス。

以下は、IP 管理プロセスに関する筆者の推測である。

オープンソースをベースにビジネスをする場合、ライセンスの汚染⁶や第三者特許が混入していると、ビジネスが成立しなくなってしまう。Eclipse の場合は IP 管理プロセスが整備されており、汚染されないような仕掛けが施されている。これが、Webtide に安心感を与えるのだろうと思う。

A.2.6 Protecode

インタビューイ CTO

ビジネス ソースコードの由来を検出ツールを販売。ソフトウェア開発において、なんというライセンスのオープンソースのコードが入っているのかを管理することができる。

動機/きっかけ Sun が提供するオープンソースの開発環境 NetBeans と比べて単純に Eclipse の方が市場が大きい。ソースコードを寄贈して Eclipse プロジェクトをリードする予定はない。

このようなツールの市場は、大手の参入もなく数社のベンチャー企業が存在している、黎明期にある。したがって、Eclipse プロジェクトを立ち上げて大手に対抗する必要もない。コモディティ化した技術ではないので、公開して他者からのイノベーションを誘発するよりも、まだまだ自社で研究開発する余地がある技術である。したがって、単純に市場を求めて Eclipse 陣営に参加しただけのようである。

A.2.7 Meristic

インタビューイ President

ビジネス 2003 年創立の米国ベンチャー。電子認証基盤の製品・サービスでビジネス。

⁶GPL にはソースコード強制公開の条項があるため、GPL ライセンスのソースコードが間違えて製品に混入しているとソース公開の義務が生じてしまう。

動機/きっかけ Eclipse の Higgins プロジェクト⁷をリードし、認証フレームワークを開発。
参加の理由は、次の3点:

- EPL ライセンスがビジネス向きであったこと
- IP 管理プロセスが整備されており安心できたこと
Apache プロジェクトも検討したが、ボランティアベースであったため不安を感じた。
- Eclipse の知名度が上がってきていたこと

A.2.8 Actuate

インタビューイ Vice President of Product Management

ビジネス 1993 年創立の米国ベンチャー。ビジネスインテリジェンス、レポートینگ製品の販売、カスタム化、サポート、トレーニングでビジネス。

動機/きっかけ Actuate はレポートینگフレームワークの Eclipse プロジェクト BIRT にソースを寄贈しリードしている。オープンソース化にあたっては次の2つの目的があった。

- 競合の市場を奪う
- 市場自体を広げる

Actuate は競合を2つのレベルに分けていた。ひとつはコグノス社のような大手、もうひとつはオープンソースのような小さなベンダー。オープンソース化によって、大手の市場に食い込むと、同時に弱小ベンダーの市場を取り込んでしまうことを狙っていた。オープンソース化によって売上が急上昇し、結果として成功した。

A.2.9 Eclipse 財団

EclipseCon 2009 での Eclipse 財団のプレゼンテーション資料 (New Member Jumpstart) より抜粋。

インタビューイ Director, Ecosystem Development

動機/きっかけ Eclipse 財団が想定する参加理由

- メンバーになる理由

⁷<http://www.eclipse.org/higgins/>

- メンバーのうち 1/3 – 1/2 は、それまで数万ドルかかっていた開発環境の費用が無料になり、浮いた分を寄付したいため
- IP 管理プロセスが整備されているから
- Eclipse ブランド (ロゴ) を利用できるから
- EclipseCon などのイベントでプレゼンスを向上できるから
- よい人材を雇用しやすいから
- 他のメンバーへの情報を配信してもらえるから
- Strategic メンバー⁸になる理由
 - Eclipse の戦略の方向性をリードできるから

A.3 まとめ

Eclipse への参加について

- Eclipse が持つユーザベース (市場) への期待。つまり、自社だけではリーチできないユーザにリーチできること、市場の創出力への期待。
- 開発コスト削減

これらは当初の想定どおりだった。

以外な発見だったのは、

- 組織や IP 管理プロセスが整備されていること

を上げるベンダーが多かったことである。良く考えてみれば、前述した 2 つは他のオープンソースプロジェクトでも期待できる事柄であるが、IP 管理プロセスまで実施しているオープンソースプロジェクトはあまり聞かない。

自社にとっての大切な知的財産たるソースコードを公開するのは非常に不安であろう。うまく行っていないオープンソースプロジェクトも多数ある。オープンソース化しても、予期せぬライセンスや特許が混入し汚染されてしまい自社では利用できないものになってしまうこともある。参加を検討しているベンダーが、このような不安を感じるであろうことは容易に想像できる。

他のオープンソースプロジェクトと比較して、Eclipse はこのような不安を軽減させる仕組みが整備されており、大切なソースコードを寄贈しても、必ず自社のメリットとなって還ってくるという安心感を与える。インタビューを通して、この安心感が、Eclipse エコシステムに参加する大きな動機であることが発見できた。

⁸Eclipse のボードミーティングへの参加資格がある

付 録 B Eclipse 参加ベンダーの変遷

次頁以降の参加ベンダーの変遷は、Eclipse.org のメンバーミーティングの資料から収集した。

Eclipse Members

#	ベンダー名	ビジネス	Eclipse Consortium					Eclipse Foundation					買収・提携	
			2001	2002	2003	2004	2005	2006	2007	2008	2009			
1	Academic Associate Member			Associates	Associates	Associates	Associates	Associates	Associates	Associates	?	x		
2	Accelerated Technology						Add-in-Provider						x	
3	Access										Add-in-Provider	Solutions		
4	AccuRev						Add-in-Provider	Add-in-Provider				x		
5	ACM Queue						Associates	Associates	Associates	Associates	Associates	Associates		
6	Active Grid							Add-in-Provider	Add-in-Provider	?		x		
7	actuate						Strategic	Strategic	Strategic	Strategic	Strategic	Strategic		
8	Acucorp	COBOL製品				Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	x	x			2007年Micro Focusが買収
9	Adacode							Add-in-Provider	Add-in-Provider	Add-in-Provider	Solutions			
10	Addison Wesley						Associates	Associates	Associates	Associates	Associates	Associates		
11	Adobe							Add-in-Provider	Add-in-Provider	Add-in-Provider	Solutions			
12	Advanced Systems Concepts			Board		Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	x			
13	Agitar Software	品質管理ソフト				Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	?	x			
14	Aldon	ソフト開発管理ツール				Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	?	x			
15	AlerPoint							Add-in-Provider	Add-in-Provider	?	x			
16	AltoWeb			Board							x			
17	AMD							Add-in-Provider	Add-in-Provider	x				
18	ANCIT							Add-in-Provider	Add-in-Provider	Add-in-Provider	Solutions			
19	andrena objects ag									Add-in-Provider	Solutions			
20	Anywhere Technologies							Add-in-Provider	Add-in-Provider	Add-in-Provider	Solutions			
21	AOL										Solutions			
22	Aonix	リアルタイムJava				Add-in-Provider	Add-in-Provider	?	?	?	x			
23	Aptana							Add-in-Provider	Add-in-Provider	Add-in-Provider	Solutions			
24	ARM							Add-in-Provider	Add-in-Provider	Add-in-Provider	Solutions			
25	ARS Computer and Consulting GmbH										Solutions			
26	Atmel									Add-in-Provider	Associates			
27	AvantSoft					Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	Solutions			
28	Band XI							Add-in-Provider	Add-in-Provider	Add-in-Provider	Solutions			
29	BEA						Strategic	Strategic	x	x	x			2008年RedHatが買
30	blackduck						Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	Solutions			
31	BLU AGE									Add-in-Provider	Solutions			
32	bluenog										Solutions			
33	BRE DEX								Add-in-Provider	Add-in-Provider	Solutions			
34	Borland	開発ツール	Board	Board	Board	Add-in-Provider	Strategic	Strategic	Strategic	x	x			
35	Brocade								Add-in-Provider	Add-in-Provider	Solutions			
36	brox								Strategic	Strategic	Solutions			
37	bsi										Solutions			
38	BuildForge						Add-in-Provider	x	x	x	x			2006年IBMが買収
39	Business	BIツール(仏)					Add-in-Provider	Add-in-Provider	Add-in-Provider	x	x			2008年SAPが買収
40	BZ Media						Associates	Associates	Associates	Associates	Associates			
41	Cape Clear Software	ESB(SOA)					Add-in-Provider	Add-in-Provider	Add-in-Provider	x	x			2008年Workdayが買収
42	CanyonBlue	UML		Board	Add-in-Provider	?	?	?	?	?	x			
43	Carleton Univ.							Associates	Associates	Associates	Associates			
44	Carnegie Mellon Univ.										Associates			
45	Catalyst Svstems	組込み	Board	Board	Add-in-Provider	Add-in-Provider	Add-in-Provider	?	?	?	x			
46	CAS										Add-in-Provider	Solutions		
47	CEA List										Associates	Associates		
48	cenit										Add-in-Provider	Solutions		
49	Centrum voor Wiskunde en Informatica										Associates	Associates		
50	CISCO										Add-in-Provider	Solutions		
51	Clemson Univ.									Associates	Associates			
52	Cloudsmith							?	Strategic	Strategic	Strategic			
53	Codign Software							Add-in-Provider	Add-in-Provider	?	x			
54	Cognos	BIツール					Add-in-Provider	Add-in-Provider	Add-in-Provider	x	x			2007年IBMが買収
55	CollabNet					Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	Solutions			
56	Communications and Media Arts						Associates	Associates	Associates	Associates	Associates			
57	Compeople								Add-in-Provider	Add-in-Provider	Solutions			
58	Computer Associates	管理ソフト				Add-in-Provider	Strategic	Strategic	Strategic	Strategic	Strategic			
59	Compuware					Add-in-Provider	Strategic	Strategic	Add-in-Provider	Add-in-Provider	Solutions			
60	Conselleria de Infraestructuras y Transporte										Associates	x		
61	Cuifl										Add-in-Provider	Solutions		
62	CWI								Associates	Associates	Associates			
63	DataMirror	リアルタイムデータ処理					Add-in-Provider	Add-in-Provider	Add-in-Provider	x	x			2007年IBMが買収
64	DevZuz								Strategic	?	x			
65	DDCI						Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	Solutions			
66	DFKI										Associates	Associates		
67	Discovery machine						Add-in-Provider	Add-in-Provider	Add-in-Provider	Add-in-Provider	x			
68	DSDM Consortium								Associates	Associates	Associates	Associates		
69	Dzone										Associates	Associates		
70	EADS							Add-in-Provider	Add-in-Provider	Associates	Associates			
71	Eclipse Plug-in Central(EPIC)						Associates	Associates	Associates	Associates	x			
72	e forum										Associates	Associates		