

異なる教義を持つ宗教間のバランス

筑波大学大学院 ビジネス科学研究科
経営システム科学専攻 修士1年 小林正人

1 はじめに

本報告は、Linux に代表されるオープンソースによるコミュニティと、Microsoft のような商業ソフトウェアのユーザ集団をモデル化し、社会の中でそれぞれのグループがどのような住み分けがなされるかについてのシミュレーションを行った結果である。

Linux に代表されるオープンソース運動は一時のブームは過ぎたとはいえ、そのコミュニティは日本においても十分に受け入れられ、定着したように見える。しかしながら、Linux が普及したとはいえ一般の PC ユーザの大多数は Microsoft 製品ユーザであり、そのシェアの絶対的な優位性はここしばらくは揺るぎそうに無いように思われる。このような条件下では、たとえ一度 Linux のようなオープンソース OS のユーザになったとしても、小数派に属することを恐れるという理由だけで再び Microsoft OS ユーザに戻る者も少なくない。特に職場の環境が Microsoft 製品である場合には、個人の価値観や主義主張とは無関係に、いやおうなく Microsoft 製品ユーザにならざるを得ないことも多い。

Linux と Microsoft OS のユーザ数の動きに関してみると、現在の状況は比較的平衡しているように見えるが、上記のような理由からユーザが突然雪崩的に多数派に移動する可能性も考えられる。そもそも、基本的な考えが異なり互換性がない2つのソフトウェアのユーザグループが、バランスを保ちながら共存することは可能なのだろうか。もしも可能だとしたらそのようなバランスの条件はどのようなものであるのか、ということはこのエージェントシミュレーション実験で調べてみたい。

ところで、相対するソフトウェアユーザ同志がお互いのツールの優越性を主張しあう様子を、しばしば「宗教戦争」という言葉で表現する。本報告のタイトルはそれを模して、オープンソースのコミュニティと Microsoft ユーザをそれぞれ異なる教義を持つ宗教の信者としてたまためたものである。

オープンソース信者は自らの成果を広く世の中に与え、それに共感する者が新たな信者として加わるのを待つという、いわば寛容な教義を持っているのに対して、Microsoft 信者は、勢力拡大のために徒党を組んで異教徒を排除するという非常に厳しく排他的な教義を持っているものとしてモデル化する。

以下では、これらの集団のモデル化と、その集団に属するメンバー同志が有限の場で自らの集団の教義にしたがって行動した場合のシミュレーション結果、そしてその結果についての考察を示す。

2 モデルの構造とルール

このモデルでは、行動原理が異なる2つの集団「OpenSource_Users」と「Microsoft_Users」に属する2種類のエージェントで構成された社会を考える。また、場の各点は「環境情報量」という値を持っている。

2-1 変数

表 2-1 モデル全体の変数

視野	各エージェントが影響を与える(受ける)範囲	1 - 3
移動速度	各エージェントの移動速度	0.5 - 2
世代交替	エージェントの入れ換えを行うかどうか	True/False
OS ランダム移動	OpenSource_Users の移動タイプ	True:ランダムに移動 False:環境情報最大値が大きい場所に移動
MS ランダム移動	Microsoft_Users の移動タイプ	True:ランダムに移動 False:環境情報量が大きい場所に移動
環境情報最大値	場の各点が持てる環境情報量の最大値	表 2-3 表 2-4

表 2-2 エージェントの内部状態

X、Y	位置
寿命	エージェントの寿命 (ステップ数)
年齢	エージェントの年齢 (ステップ数)
スキル	Microsoft_Users の持つ情報量
ダメージ	OpenSource_Users が攻撃された量

環境情報最大値に関しては表 3-1 のデータを使用した。また、比較のため表 3-2 のデータを用いたシミュレーション実行も行った。

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	2	3	4	5	6	8	6	5	4	3	2	0	0	0	0
0	0	0	0	0	3	4	5	6	8	10	8	6	5	4	3	0	0	0	0
0	0	0	0	0	4	5	6	8	10	12	10	8	6	5	4	0	0	0	0
0	0	0	0	0	5	6	8	10	12	14	12	10	8	6	5	0	0	0	0
0	0	0	0	0	6	8	10	12	14	17	14	12	10	8	6	0	0	0	0
0	0	0	0	0	8	10	12	14	17	20	17	14	12	10	8	0	0	0	0
0	0	0	0	0	6	8	10	12	14	17	14	12	10	8	6	0	0	0	0
0	0	0	0	0	5	6	8	10	12	14	12	10	8	6	5	0	0	0	0
0	0	0	0	0	4	5	6	8	10	12	10	8	6	5	4	0	0	0	0
0	0	0	0	0	3	4	5	6	8	10	8	6	5	4	3	0	0	0	0
0	0	0	0	0	2	3	4	5	6	8	6	5	4	3	2	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 2-3 環境情報最大值 (1)

表 2-4 環境情報最大値 (2)

0	0	0	0	0	1	1	2	3	4	5	4	3	2	1	1	0	0	0	0
0	0	0	0	1	1	2	3	4	5	6	5	4	3	2	1	1	0	0	0
0	0	0	1	1	2	3	4	5	6	7	6	5	4	3	2	1	1	0	0
0	0	1	1	2	3	4	5	6	7	8	7	6	5	4	3	2	1	1	0
0	1	1	2	3	4	5	6	7	8	9	8	7	6	5	4	3	2	1	1
1	1	2	3	4	5	6	7	8	9	11	9	8	7	6	5	4	3	2	1
1	2	3	4	5	6	7	8	9	11	12	11	9	8	7	6	5	4	3	2
2	3	4	5	6	7	8	9	11	12	14	12	11	9	8	7	6	5	4	3
3	4	5	6	7	8	9	11	12	14	16	14	12	11	9	8	7	6	5	4
4	5	6	7	8	9	11	12	14	16	18	16	14	12	11	9	8	7	6	5
5	6	7	8	9	11	12	14	16	18	20	18	16	14	12	11	9	8	7	6
4	5	6	7	8	9	11	12	14	16	18	16	14	12	11	9	8	7	6	5
3	4	5	6	7	8	9	11	12	14	16	14	12	11	9	8	7	6	5	4
2	3	4	5	6	7	8	9	11	12	14	12	11	9	8	7	6	5	4	3
1	2	3	4	5	6	7	8	9	11	12	11	9	8	7	6	5	4	3	2
1	1	2	3	4	5	6	7	8	9	11	9	8	7	6	5	4	3	2	1
0	1	1	2	3	4	5	6	7	8	9	8	7	6	5	4	3	2	1	1
0	0	1	1	2	3	4	5	6	7	8	7	6	5	4	3	2	1	1	0
0	0	0	1	1	2	3	4	5	6	7	6	5	4	3	2	1	1	0	0
0	0	0	0	1	1	2	3	4	5	6	5	4	3	2	1	1	0	0	0

環境情報量はその場所において得ることができる情報の量を表している。この量は Microsoft_Users エージェントのスキル値を上昇させ、OpenSource_Users エージェントへ変化させる働きをもつ。

環境情報最大値は、その場所が最大どれだけの環境情報量を持つことができるかを規定する。例えば全社で Microsoft 製品を使用することを義務付けているような場所では環境情報最大値は 0 となる。

表 2-3、表 2-4 はそれぞれ中央に環境情報最大値の山があるが、表 2-3 の山は非常に険しい形をしており、周辺部には環境情報最大値 0 の平野が広がっている。それに対して表 2-4 の山はなだらかで、裾野部分にも 0 になる部分は少ない。

2-2 ルール

ステップ毎に、環境情報量は1ずつ減少していく。ただし0未満にはならない。
各集団に属するエージェントの基本的行動ルールは以下の通りである

OpenSource_Users

- ・OS ランダム移動が True の場合はランダムに移動し、False の場合は環境情報最大値が大きい点に向かって移動する。
- ・自分のいる点を中心として、視野で設定された範囲内の環境情報量を環境情報最大値まで引き上げる。
- ・視野で設定された範囲内に OpenSource_Users エージェントがいれば、その人数分だけダメージが回復する。

Microsoft_Users

- ・MS ランダム移動が True の場合はランダムに移動し、False の場合は環境情報量が大きい点に向かって移動する。
- ・移動した場所の環境情報量が自分のスキルに加わる。スキルが一定以上の値になると、OpenSource_Users へと変化する。
- ・自分のいる点を中心として視野で設定された範囲内にいる OpenSource_Users エージェントを攻撃しダメージを与える。攻撃の量は視野内にいる Microsoft_Users エージェント数に比例して増加する。すなわち OpenSource_Users エージェント側からみると視野内の Microsoft_Users エージェント数の2乗に比例した量のダメージを受けることになる。

(与えられたダメージが一定量を越えると OpenSource_Users エージェントはこの時点で消滅する)

- ・視野で設定された範囲内に Microsoft_Users エージェントがいれば、その人数分だけスキルが減少する。

上記の基本ルールに加え、移動速度、各エージェント数の初期値を設定することができる。

また、世代交替を True にすると各エージェントは 20 から 60 ステップの寿命の後に消滅する。この場合エージェント同志は以下のルールで子供を作る

- ・自分自身の年齢が 18 ステップ以上の場合、6%の確率で視野 1 以内のエージェントを探す。

- ・この時周囲に年齢が 18 ステップ以上の同種のエージェントがいる場合同種の子供ができる。ただし、10%の確率で年齢 18 ステップ以上の別種のエージェントを探し、子供をつくる。この子供が属するグループは 50%の確率で決定する。
- ・いずれの場合も新しくできた子供はランダムな位置に置かれる。

3 実行結果

各条件を変えて行ったシミュレーション結果のうち、特徴的なものについて各エージェント数の時系列グラフを示す

グラフは、青い線が OpenSource_Users エージェント数、赤い線が Microsoft_Users エージェント数を表している。

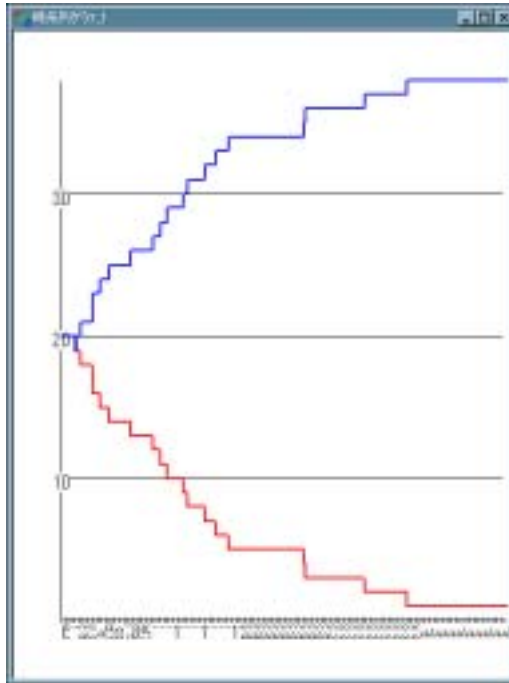


図 3-1 20 個 20 個 移動速度 1 MS ランダム移動 視野 1 世代交代なし

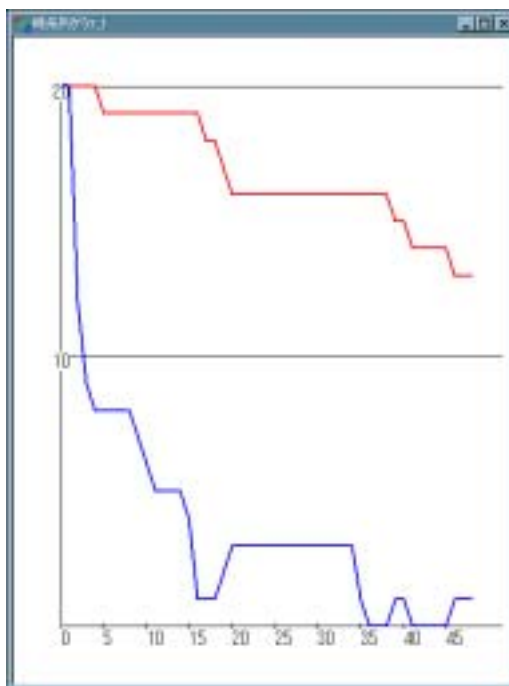


図 3-2 20 個 20 個 移動速度 1 MS ランダム移動 視野 3 世代交代なし

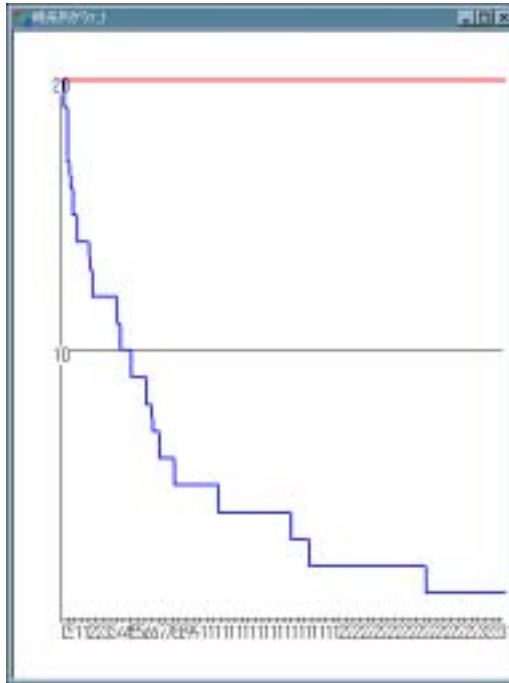


図 3-3 20 個 20 個 移動速度 1 MS/OS ランダム移動 視野 1 世代交代なし

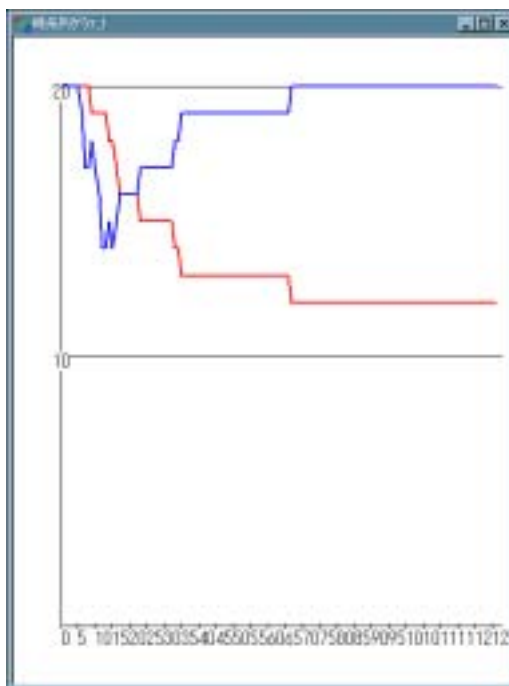


図 3-4 20 個 20 個 移動速度 1 ランダム移動なし 視野 1 世代交代なし



図 3-5 20 個 20 個 移動速度 1 ランダム移動なし 視野 2 世代交代なし

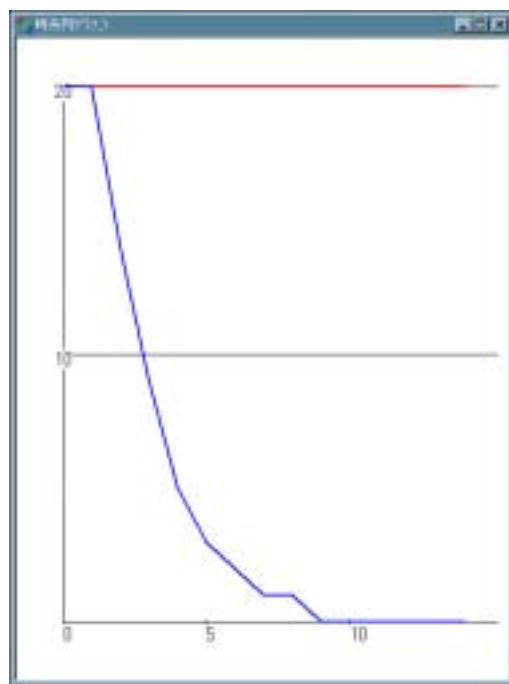


図 3-6 20 個 20 個 移動速度 2 ランダム移動なし 視野 3 世代交代なし

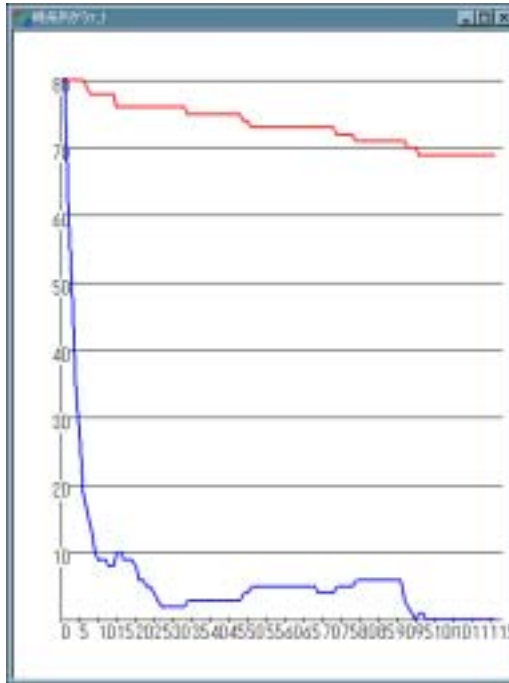


図 3-7 80 個 80 個 移動速度 1 MS ランダム移動 視野 1 世代交代なし

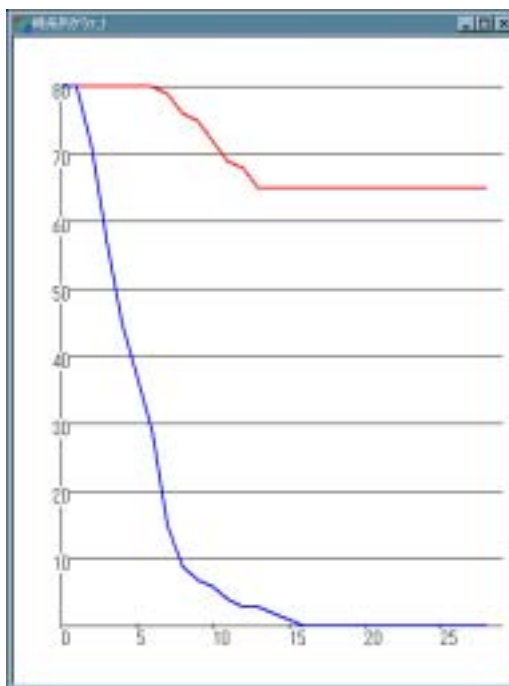


図 3-8 80 個 80 個 移動速度 1 ランダム移動なし 視野 1 世代交代なし

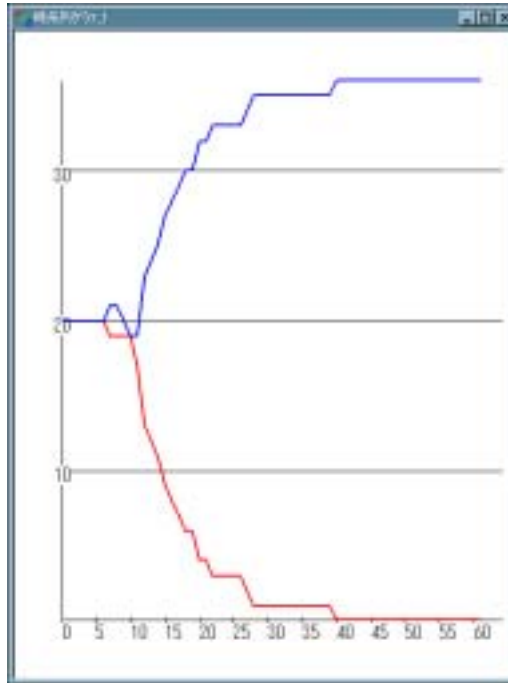


図 3-9 20個 20個 移動速度 1 ランダム移動なし 視野 1 世代交代なし
(環境情報最大値データ 2)

4 考察

OpenSource_Users、Microsoft_Users 双方のエージェントの行動ルールは非対称である。特に OpenSource_Users エージェントは消滅するだけなのに対して Microsoft_Users エージェントは OpenSource_Users エージェントに変化するというルールは、OpenSource_Users に有利に働くように見える。それにもかかわらず、多くの場合 OpenSource_Users エージェントは全滅することが多い。

特に、図 3-7、図 3-8 に見られるように、開始時に双方のエージェント数が多数であった場合、その他のパラメータをどのように変化させても OpenSource_Users が生き残ることはできなかった。

また、自らの移動をランダムにすること、視野を広くすることも OpenSource_Users に対して不利な条件になることが多いということもわかる。

2つの集団が共存しているのは図 3-4、図 3-5 だが、この条件は

- ・ 開始時の個体数が少ない
- ・ 視野が狭い
- ・ ランダムな動きをしない
- ・ 移動速度が遅い

というものである。

図 3-6 を見るとわかるように、視野を広くして移動速度を上げるとかなり早い時期に OpenSource_Users エージェントは全滅してしまう。これはそれぞれの条件を個別に変化させた場合でも、全滅までのステップ数が変化するだけで結局は全滅してしまう。

また図 3-9 は、図 3-4 と同じ条件で環境情報最大値のデータを表 2-4 に変えただけだが、ここでは共存はおこらず Microsoft_Users の方が全滅してしまっていることがわかる。

以上のことを、単純に現実世界の条件に当てはめて考えると、最終的に得られる OpenSource_Users と Microsoft_Users の共存条件は

- ・ **豊かな環境になる可能性がある場所は限られていて、多くの劣悪な環境があることが不可欠**
- ・ **OpenSource_Users は豊かな環境になる可能性がある場所を探し、そこで仕事をする**
- ・ **Microsoft_Users は OpenSource_Users の作成物を利用するために探しまわる**
- ・ **視野は狭い**
- ・ **移動(検索)速度は遅い**

ということになる。

これらの条件は、少なくとも今現在の世界に存在する現実の OpenSource_Users と Microsoft_Users に適用することは不可能であるといってよい。

また、今回示さなかったが、世代交代を True にすると OpenSource_Users は常に全滅し、それ以外にも多くの条件下でのシミュレーションにおいて OpenSource_Users が全滅することが多いということも、2つの集団の共存という目的に関しては悲観的な結果に終わったといえる。

この結果の原因としては、OpenSource_Users と Microsoft_Users の行動ルールのモデル化が現実と合っていなかったためと考えられる。しかしながら逆に、このモデルに近い行動ルールにしたがって行動する2つの集団が仮に存在するとしたら、最終的に得られた共存条件には興味深いものがあるといえるだろう。

以上