

Title	通信事業者における、B2CからB2B2Cへの販売モデル変更による影響分析： エージェントベースモデルを利用したシミュレーションによるアプローチ
Sub Title	
Author	神立, 佳広(Kandatsu, Yoshihiro) 高橋, 大志(Takahashi, Hiroshi)
Publisher	慶應義塾大学大学院経営管理研究科
Publication year	2016
Jtitle	
JaLC DOI	
Abstract	
Notes	修士学位論文. 2016年度経営学 第3150号
Genre	Thesis or Dissertation
URL	https://koara.lib.keio.ac.jp/xoonips/modules/xoonips/detail.php?koara_id=KO40003001-00002016-3150

慶應義塾大学学術情報リポジトリ(KOARA)に掲載されているコンテンツの著作権は、それぞれの著作者、学会または出版社/発行者に帰属し、その権利は著作権法によって保護されています。引用にあたっては、著作権法を遵守してご利用ください。

The copyrights of content available on the Keio Associated Repository of Academic resources (KOARA) belong to the respective authors, academic societies, or publishers/issuers, and these rights are protected by the Japanese Copyright Act. When quoting the content, please follow the Japanese copyright act.

慶應義塾大学大学院経営管理研究科修士課程

学位論文（ 2016 年度）

論文題名

通信事業者における、B2C から B2B2C への販売モデル変更による影響分析
—エージェントベースモデルを利用したシミュレーションによるアプローチ—

主 査	高橋 大志
副 査	小林 喜一郎
副 査	坂下 玄哲
副 査	

氏 名	神立 佳広
-----	-------

論文要旨

所属ゼミ	高橋 大志 研究会	氏名	神立 佳広
(論文題名)			
<p>通信事業者における、B2C から B2B2C への販売モデル変更による影響分析 –エージェントベースモデルを利用したシミュレーションによるアプローチ–</p>			
(内容の要旨)			
<p>◆はじめに</p> <p>2015 年から、固定通信サービス市場においては、大きな変化が生じている。従来、数社からのみ提供され、寡占市場の固定型光回線サービスであったが、様々な企業が自社のサービスを展開し始めた。その結果、2014 年まで数社であった市場のプレイヤーが、現在では提供会社は 100 社近くに上っており競争が激化している。各社がサービスを提供開始した背景には、既存プレイヤーの動きが大きく影響している。まず、大きな影響を与えたのが、2012 年に KDDI が価格競争を始めたことである。KDDI やケーブルテレビを始めとした企業は 2012 年から、携帯電話とのセット割を始めとした価格による訴求により、NTT 東西に対抗し始めた。次に、顧客の乗換周期の変化と新規技術の発生という 2 つの外部環境の変化である。KDDI の価格競争をきっかけとし、顧客が回線サービスの切り替えの検討を行うようになった。更に、携帯と固定通信、更には電気など多様なセット販売を行うようになり、顧客の乗換周期が多様化してきている。一方、固定通信市場以外に目を向けると、IOT やクラウド技術、仮想化技術等の新たな技術の発達により固定通信の安定した通信技術のニーズが増加してきている。これらの変化を元に、NTT 東西は、2015 年、従来の直接販売 (B2C) から卸販売 (B2B2C) 事業へとシフトし、卸売り先の企業が様々な名前を付けたサービスを開始した。</p> <p>◆検証</p> <p>エージェントベースモデルを通じ、疑似的に収益性のシミュレーションを実施。従来の販売モデルを続けたケースと新しいモデルを続けたケースをシミュレーションし比較することにより、今回の販売モデルの変更が正しい判断であったのかを探っていった。尚、本研究では具体的に下記 3 条件の検証を実施した。</p> <ul style="list-style-type: none"> ・外部変化がない環境での収益性比較 ・顧客の乗換周期が変化する環境での収益性比較 ・市場の顧客が増加する環境での収益性比較 <p>◆検証結果</p> <ul style="list-style-type: none"> ・外部変化がない環境での収益性比較 <p>新モデルに切り替えることにより競合との価格競争が可能となりユーザ数を多く獲得することが可能となる。その一方で、同じ NTT ユーザではあるが単価が低い卸売り販売への流入が発生するため収益性は低くなった。</p> ・顧客の乗換周期が変化する環境での収益性比較 <p>顧客の乗換周期が短くなるほど NTT が多くユーザを獲得できるようになる。一方で、顧客の乗換に合わせ価格競争がさらに激化するため収益性は周期が短くなるほど低くなった。</p> ・市場の顧客が増加する環境での収益性比較 <p>本研究にて設定した環境下では、市場の顧客増加がある程度見込まれた場合、新モデルのビジネスモデルが旧モデルの収益性を上回ることが確認できた。ただし、新たなビジネスが生まれ新たな顧客が増加しなければ旧モデルの方が収益性が高いことがわかった。</p> <p>◆考察</p> <p>本ビジネスモデル変更を成功に導くためには、今後、外部環境の変化を機会としてとらえ、自らの収益を高めるために市場へアプローチをしていくことが重要である。まず、顧客の乗換周期を長くすることである。2 年縛り等顧客の見直しを遅くすることにより収益性を上げることが可能である。次に、新たな顧客を生み出すことである。今回のビジネスモデルの変更により、新たな企業との協業を可能となり、新たなビジネスを生み出す体制が整った。今後 IOT やクラウド化等、トレンドの技術を組み合わせ卸売り企業の新たなビジネスを生み出すサポートをすることにより、新たな顧客を生み出していくことが重要である。</p>			

目次

1. はじめに.....	2
2. 先行研究.....	3
3. 目的.....	4
4. 分析手法.....	4
5. モデル.....	5
5-1. 提供形態.....	5
5-2. 分析の前提.....	6
5-3. モデル詳細.....	7
6. 検証.....	8
6-1. 検証の流れ.....	8
6-2. 検証詳細.....	8
7. 検証結果.....	11
7-1. 検証①：外部変化なしの場合の検証.....	11
7-2. 検証②：顧客の乗換え周期が変わった場合の検証.....	20
7-3. 検証③：市場に新規顧客が増加する場合の検証.....	38
8. 考察.....	38
9. 本研究の限界と今後の課題.....	39
10. 参考文献.....	39
11. 付属資料.....	40
12. 謝辞.....	52

1. はじめに

2015年から、固定通信サービス市場においては、大きな変化が生じている。従来、数社からのみ提供され、寡占市場の固定型光回線サービスであったが、株式会社ソフトバンクや株式会社NTTコミュニケーションズ等、様々な企業が自社のサービスを展開し始めた。その結果、2014年まで数社であった市場のプレイヤーが、現在では提供会社は100社以上に上っており競争が激化している。各社が光回線サービスを提供開始した背景には、既存プレイヤーであった、東日本電信電話株式会社(以下NTT東)、西日本電信電話株式会社(以下NTT西)とKDDI株式会社(以下KDDI)の動きが大きく影響している。

まず、大きな影響を与えたのが、2012年にKDDIが価格競争を始めたことである。近年、国内の通信事業は成熟市場となっており、特に各家庭にサービスを行う固定型インターネットサービスは携帯電話等を始めとした無線通信の普及により、今後の縮小が予想されている。また、固定型インターネットサービス市場の中を見ると、NTT東、NTT西(以下NTT東西)が光回線で7割程度の圧倒的なシェアを取っているが、KDDIやケーブルテレビを始めとした企業は、携帯電話とのセット割を始めとした価格による訴求により、NTT東西に対抗し始めた。

次に、顧客の乗換周期の変化と新規技術の発生という2つの外部環境の変化である。KDDIの価格競争をきっかけとし、顧客が回線サービスの切り替えの検討を行うようになった。更に、携帯と固定通信、更には電気など多様なセット販売を行うようになり、顧客の乗換周期が多様化してきている。一方、固定通信市場以外に目を向けると、IoTやクラウド技術、仮想化技術等の新たな技術の発達により固定通信の安定した通信技術のニーズが増加してきている。従来は伸びが鈍化していた市場であるが、今後の戦略次第で更なる伸びを目指せる市場となりつつある。以上の競合の動きと将来起こるであろう2点の外部環境の変化より、NTT東西は戦略の変更を選択したと言える。

一方、NTT東西が卸売り事業を選択した背景としては、NTT東西の成り立ちから生じている特殊な制約の影響が大きい。NTT東西は、旧逓信省であり、電電公社を経て1984年に民営化された企業である。今回の各社光サービスの提供開始には、民営化された際の日本電信電話株式会社等に関する法律(以下NTT法)が大きく影響をしている。NTT法では、NTT東西は、“第3条 あまねく日本全国における適切、公平かつ安定的な供給の確保に寄与する”と定められている。つまり、特定の企業のみと提携することは制限されており、すべての企業を同等に扱わなくてはならないと規定されている。本法律により、KDDIやケーブルテレビが行っている、NTTドコモとの携帯電話とのセット割を提供することが出来なかった。そこで、NTT東西は、2015年、KDDIやケーブルテレビへの対抗と

して、図1のように、従来の直接販売(B2C)から卸販売(B2B2C)事業へと販売方法をシフトし、同様のセット割サービスを開始した。卸販売は、ソフトバンクやNTTコミュニケーションズ等、卸売りを希望する企業すべてに自社の光回線を販売することであり、様々な名前の付いた光回線サービスが乱立することとなった。

本研究では、NTT東西が実施した、販売モデルの変更について、その背景を踏まえたうえで、意思決定の妥当性について考察を行っていく。

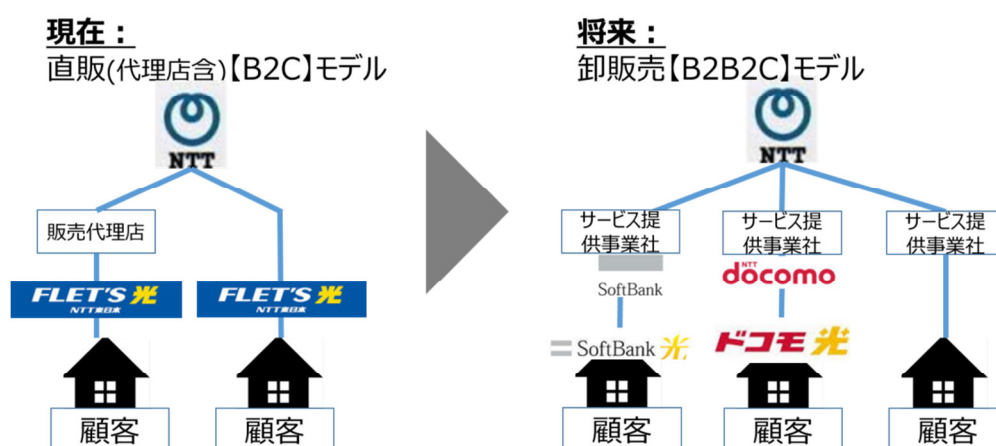


図1 旧モデルと新モデルの比較

2. 先行研究

M. E. ポーター(1995)によると、衰退期にある市場の戦略は4つある。1つ目はシェアを獲得し業界唯一の生き残りを目指す“リーダーシップ戦略”、2つ目は事業から回収する現金を最大化する“刈り取り戦略”、3つ目は高い収益性のセグメントの防衛に力を入れる“拠点確保戦略”、4つ目が事業を売却する“即時撤退戦略”である。4つの戦略をNTT東西に当てはめると、NTT法により事業売却や拠点を絞り込むことが規制されているため、“リーダーシップ戦略”と“刈り取り戦略”の選択肢があると言える。また、ゲイリーハメル、C・K・プラハラード(2001)によると、未来に向けて先を見て戦略を変え自分自身を変革していくことが必要であると説いており、未来への構想を有利に展開するため、企業提携が有効な手段であると述べている。

一方で、B2B2Cビジネスの代表としてはOEMの形態があるが、山田(1992)によ

ると、市場成熟期における OEM 戦略の目的としては、大量生産によるコストダウン及び早期回収が可能という点と、競争業者を減らし無意味な競争を回避する目的があると言われている。

以上の先行研究より、今回のビジネスモデルの変更は将来を見据えた正しい戦略であると読めるが、本研究では定量的に正当性を検証していきたい。

3. 目的

NTT 東西の直接販売 (B2C) から、卸売り販売 (B2B2C) へのビジネスモデルの変更について両販売モデルを比較し分析することにより、意思決定の正当性について検証を行う。卸売り販売 (B2B2C) の形態は、通信事業社としては世界初の事例であり、且つ、圧倒的なシェアを持つ直接販売 (B2C) 企業が、卸売り販売 (B2B2C) に変更を行うことも他業界を含め事例は限定的である。本意思決定を解析することにより、本ビジネスモデルの変更を今後他企業が活用する余地があるのか検証を進めたい。

4. 分析手法

本研究では、時系列で消費者の動向をシミュレーションするために、エージェントベースモデルのソフトウェアである、株式会社 NTT データ数理システム社の S4(エスクワトロ)Simulation System(以下 S4)を使用する。エージェントベースモデルは、人や組織などの自律的なエージェントの行為をシミュレートし、システム全体に与える影響の評価を行う方法である。複数のエージェントが同時に活動し、相互作用する状況をシミュレートすることによって、複雑な現象を再現し、予測することが可能である。

5. モデル

本研究では、下記モデルにて検証を進めていく。図 2 に S4 の画面を示す。

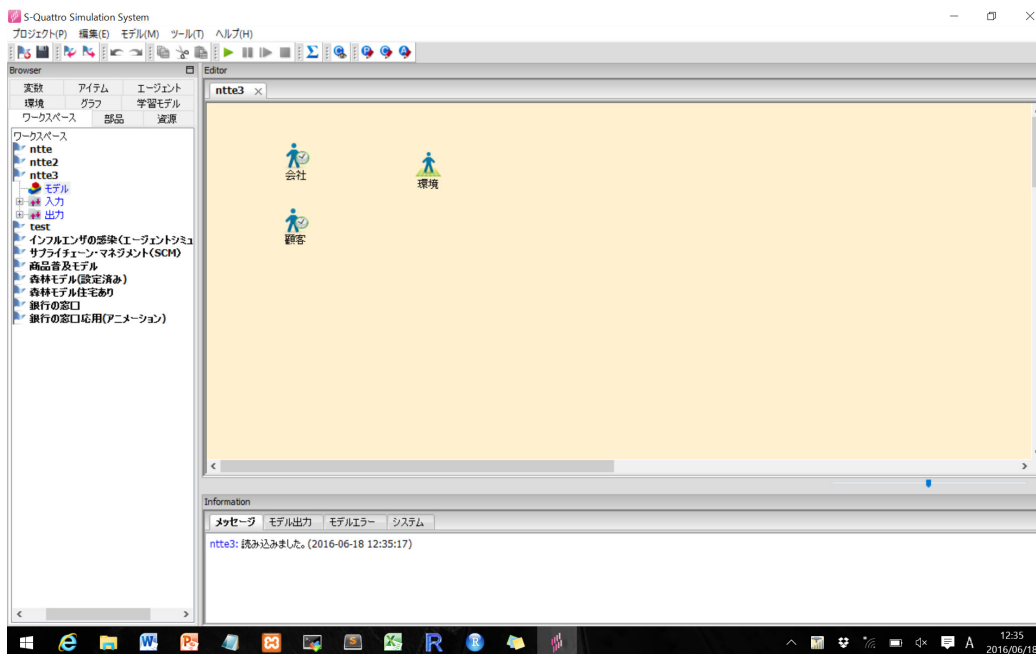


図 2 S4 シミュレーション画面

5-1. 提供形態

従来の提供形態である直接販売（B2C）モデルと現在のモデルである卸売り販売（B2B2C）モデルについて下記に違いを示す。

i) 旧モデル：直接販売（B2C）モデル

従来の NTT 東西の B2C モデルでは、NTT 東西の自社サービス“フレッツ光”として下記 2 パターンにて販売を実施。

a) 直接販売

NTT 東西の電話販売窓口である“116 コールセンター”や Web 販売窓口の“Web116”、または法人営業により直接 NTT 東西から顧客へ販売を実施する方法。

b) 代理店販売

家電量販店や電気工事会社等に NTT 東西が販売を委託し、フレッツ光を消費者へ販売する方法。本販売方法は販売委託のみのため、販売した

際にのみ手数料が発生。図3に示す通り、その後の開通工事から保守はNTT東西が請け負う。

ii) 新モデル：卸売り販売（B2B2C）モデル

ソフトバンク社やNTTドコモ社等にサービス卸を行い、各社の光サービスとして販売を実施。図3のように本提供形態では、インフラ構築、サービス提供の一部である開通工事、カスタマサポートの一部である交渉修理のみNTT東西が担当をし、その他の業務についてはサービス提供会社が担当をする。尚、NTT法により、全てのサービス提供会社に対し、NTT東西は平等な扱い(提携及び販売価格は公平性を保たなければならない)をする必要がある。

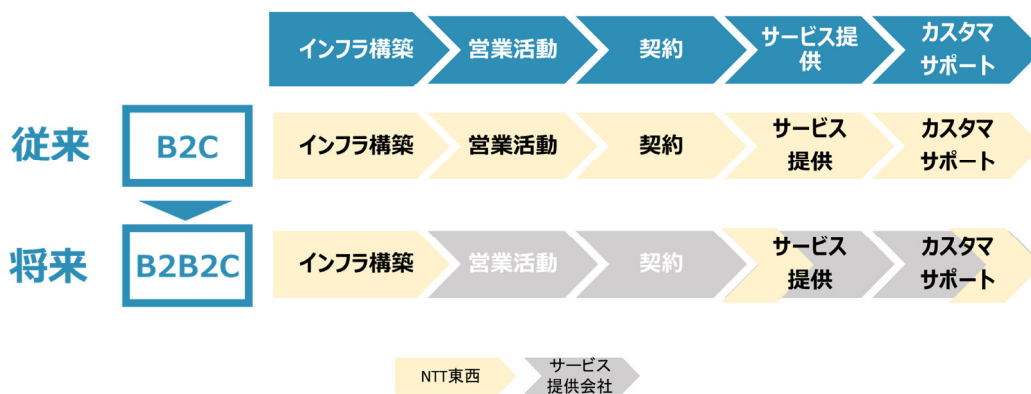


図3 旧モデルと新モデルの役割分担

5-2. 分析の前提

今回のシミュレーションにあたり、前提の環境を下記の通り定める。

- i) シミュレーション期間：10年間
 ※固定通信サービスのライフサイクルを考慮し設定
 (参考：前サービス ADSL サービス約 10 年)
- ii) 消費者世帯数の動向：飽和市場のため、新規流入は 50 万人/年で一定と設定する。
- iii) エリア：NTT 東エリア

5-3. モデル詳細

モデルの詳細を以下に示す。

5-3-1. エージェント

今回シミュレーションにおいては以下2種類のエージェントを設定する。

i) 企業エージェント

固定光通信サービスを提供する企業を企業エージェントと置く。NTT東(販売代理店を含む)、NTT東卸業者、KDDIの3種類のエージェントを設定。それぞれのエージェント数は以下の通り。

- ・NTT東：1 エージェント
- ・NTT東卸業者：9 エージェント
- ・KDDI：1 エージェント

ii) 消費者エージェント

固定光通信サービスの切り替えを選択する世帯を消費者エージェントと置く。エージェント数は今回対象のNTT東エリア光加入者数(2015年時点))である1400万世帯とする。

5-3-2. エージェントの初期値設定

各エージェントに以下の初期値を設定する。

i) 企業エージェント

実市場と条件を合わせるため、各社が獲得している消費者数と販売価格を初期値として規定する。尚、各社の消費者数は市場のシェアと合わせ、NTT東日本7割、KDDI3割(2015年時点のシェア比率)にて配分をする。

- ・NTT東エージェント
979の消費者エージェントを抱え、販売価格を4700円とする。
- ・NTT東日本卸業者エージェント
初期値では事業開始のため、消費者エージェントを0とする。納入価格は2700円で据え置き、販売価格は4700円を初期値とする。
- ・KDDI エージェント
421の消費者エージェントを抱え、販売価格は4700円を初期値とする。

ii) 消費者エージェント

特に設定なし。

5-3-3. エージェントの前提

i) 企業エージェント

NTT 東卸業者の数は9 エージェントから増減しない。

ii) 消費者エージェント

消費者の購買は価格のみに影響を受け、それ以外の広告効果等は除く。
また、消費者の価格への効用は期間中一定とし変化しないものとする。

6. 検証

項番 5 で規定したモデルを元に、両モデルの 10 年間の収益性の比較を行う。収益性については、費用等不明確な点が多いことから明確に差が生じる下記計算式にて評価する。

$$\begin{aligned} \text{収益性} &= \text{固定回線サービスでの収益総額} + \text{効率化による費用削減額} \\ &= \text{加入者当たりの収益} \times \text{NTT 加入者数} + \text{効率化による費用削減額} \\ &\quad \text{※効率化による費用削減額：1800 億円/年} \\ &\quad (\text{『通信業界の裏側が分かる 2017』日経コミュニケーションより}) \end{aligned}$$

6-1. 検証の流れ

項番 5 で規定したモデルを元に収益性に関し下記 3 点の検証を実施する。

検証①：外部変化なしの場合の検証

項番 5 で規定したモデルにより両モデルの収益性の比較を行う。

検証②：顧客の乗換え周期が変わった場合の検証

項番 6 で述べた市場の変化である顧客乗換え周期の変化を考慮し、乗換え周期が縮まった場合に両モデルの収益性に関しどのような変化が生じるか検証をする。

検証③：市場に新規顧客が増加する場合の検証

市場環境が飽和市場ではなく新規顧客増加に変化した場合の収益性に変化について検証を行う。

6-2. 検証詳細

6-2-1. 検証①：外部変化なしの場合の検証

外部変化がなく現状の条件で10年間を回した場合について両モデルをシミュレーションし、比較する。

6-2-1-1. 各エージェントの戦略

下記の通り、エージェントごとに意思決定時の戦略を設定する。尚、全体の流れは図4のフロー図に示す。

i) 企業エージェント

各企業は以下の戦略で意思決定する。尚、NTT東日本エージェントは価格改定には総務省の認可が必要となり容易に下げることが出来ないため、価格を10年間据え置くこととする。

a) NTT東日本エージェント

- ・価格を据え置く。

b) NTT東卸業者エージェント

- ・各社、定期的(1年に1回)に販売価格を見直すことができる。
- ・前2期を比べ、ユーザ数が減少している場合は価格を下げる。
- ・価格を上げることはできない。下げるか据え置くかの2択から選択する。尚、価格は2,755円が最低ラインとする(H28年度卸価格)

c) KDDI エージェント

- ・定期的(1年に1回)に販売価格を見直すことができる。
- ・前2期を比べ、ユーザ数が減少している場合は価格を下げる。
- ・価格を上げることはできない。下げるか据え置くかの2択から選択する。

ii) 消費者エージェント

消費者は以下で設定する効用関数によって意思決定を行う。

- ・消費者購買は価格のみに影響される。
- ・消費者は定期的(1年に1回)に販売店変更の検討を行う。下記効用関数にて確率を付与。

$$u(i, j) = C_i - T_{ij}$$

$$P(A) = \frac{\exp^{\hat{u}(i, j) \alpha}}{\exp^{\hat{u}(i, j) \alpha} + \exp^{\hat{u}(j, i) \alpha}}$$

$u(i, j)$: i から j へ切替えた際の効用

$P(A)$: i から j へ切替える確率

C_i : i の点数

T_{ij} : i から j に切り替える際の減点(手続き等)

α : 係数

- ・消費者の効用関数は年を経ても変化せず一定とする。

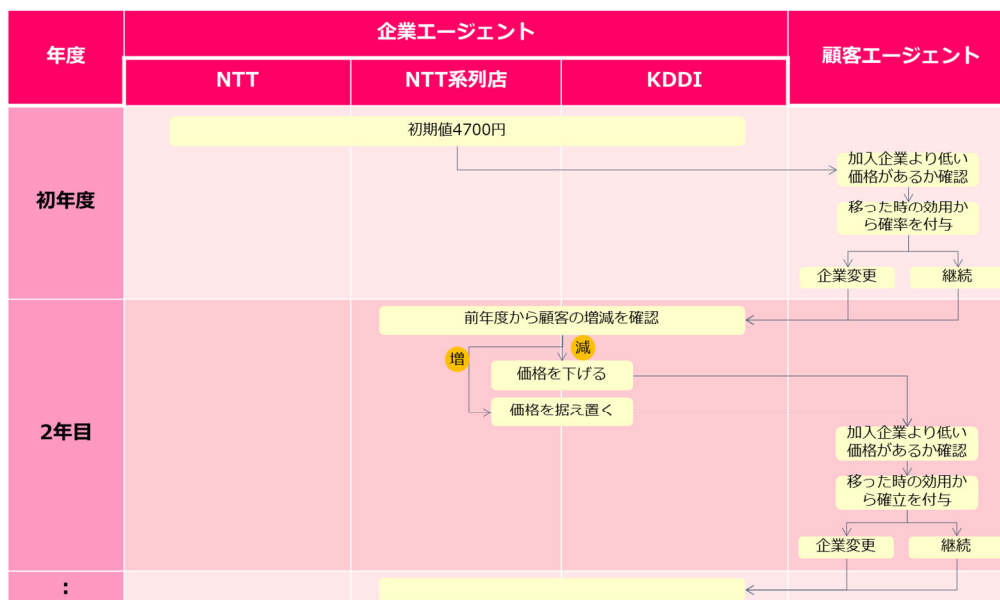


図4 エージェントのフロー図

6-2-2. 検証②：顧客の乗換え周期が変わった場合の検証

項番 6-2-1. の条件から、顧客の乗換え周期が変わったことを想定し短くなったシナリオと長くなったシナリオの2種類をシミュレーションし、両モデルの収益性の比較を行う。

6-2-2-1. 乗換え周期が長くなった場合のシナリオ

6-2-2-1-1. 各エージェントの戦略

顧客の乗換え周期が変わったことを想定し、項番 6-2-1 から下記項目を変更しモデルの収益性について比較を行う。

i) 企業エージェント

下記事項を変更する。

b) NTT 東卸業者エージェント

- ・各社の販売価格の見直しを1年に1回から2年に1回とする。

c) KDDI エージェント

- ・販売価格の見直しを1年に1回から2年に1回とする。

ii) 消費者エージェント

- ・消費者の販売店変更の検討を1年に1回から2年に1回とする。

6-2-2-2. 乗換え周期が短くなった場合のシナリオ

6-2-2-2-1. 各エージェントの戦略

顧客の乗換え周期が変わったことを想定し、項番 6-2-1 から下記項目を変更しモデルの収益性について比較を行う。

i) 企業エージェント

下記事項を変更する。

b) NTT 東卸業者エージェント

- ・各社の販売価格の見直しを1年に1回から半年に1回とする。

c) KDDI エージェント

- ・販売価格の見直しを1年に1回から半年に1回とする。

ii) 消費者エージェント

- ・消費者の販売店変更の検討を1年に1回から半年に1回とする。

6-2-3. 検証③：市場に新規顧客が増加する場合の検証

新たなビジネスが生まれ新たな顧客が増えることを想定し、顧客が初期値よりも増える場合、収益性の比較にどのような影響を及ぼすかを検証する。尚、検証方法は、項番 6-2-1 の変化なしのシナリオ、項番 6-2-2 にて検証する乗換え周期が短くなるシナリオ及び長くなるシナリオの計 3 シナリオを元に、Excel にて検証を行う。

7. 検証結果

7-1. 検証①：外部変化なしの場合の検証

7-1-1. 新モデル(B2B2C)

図 5 に各社のユーザ数の経年推移を示す。NTT 東エージェントがユーザ数を減らす一方、KDDI エージェント及び NTT 東卸売業者エージェントがユーザ数を伸ばしていることが読み取れる。

ユーザ数(万人)

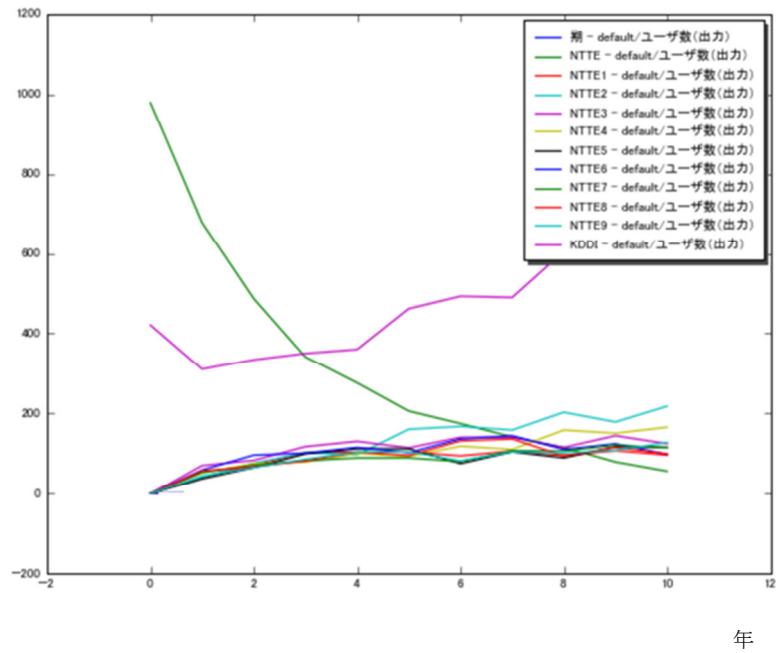


図5 各社ユーザ数の経年推移

また、図6に各社の提供価格の経年推移を示す。NTT東エージェント以外は価格競争を行い価格を年々下げていることがわかる。

提供価格(円)

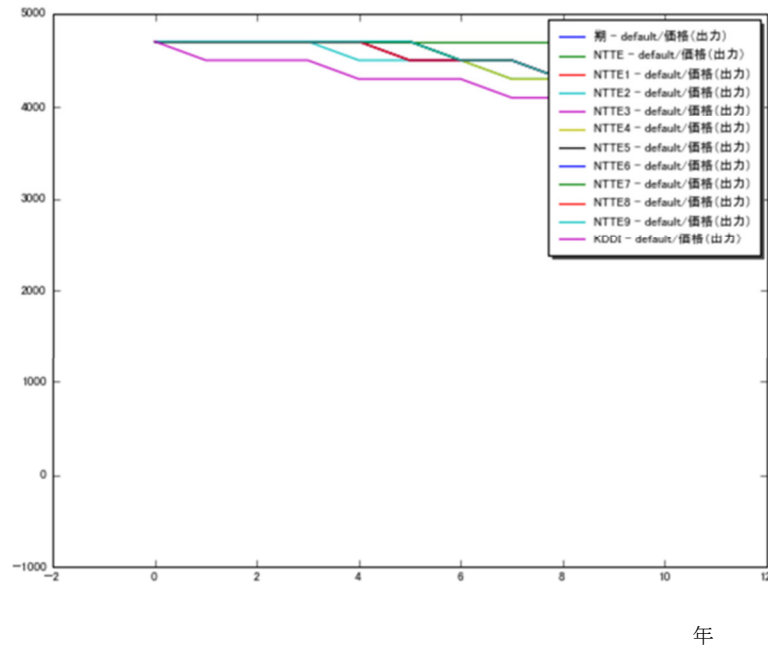


図6 各社提供価格の経年推移

7-1-2. 旧モデル(B2C)

図7に各社のユーザ数の経年推移を示す。NTT 東、KDDI エージェントともにユーザ数を伸ばすが KDDI エージェントの伸びが大きい。

ユーザ数(万人)

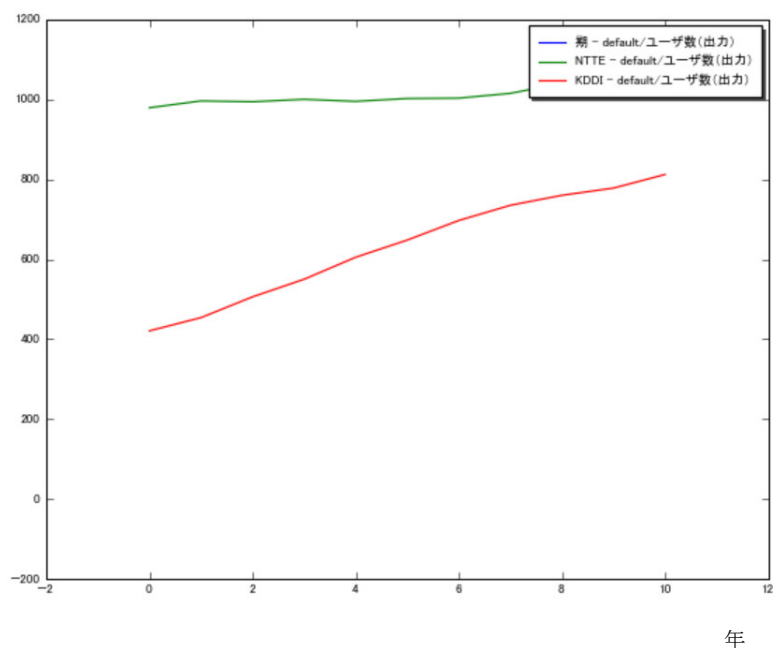


図7 各社ユーザ数経年推移

また、図8より、価格競争を行っていないことから、NTT 東、KDDI ともに価格は一定であることがわかる。

月額提供価格(円)

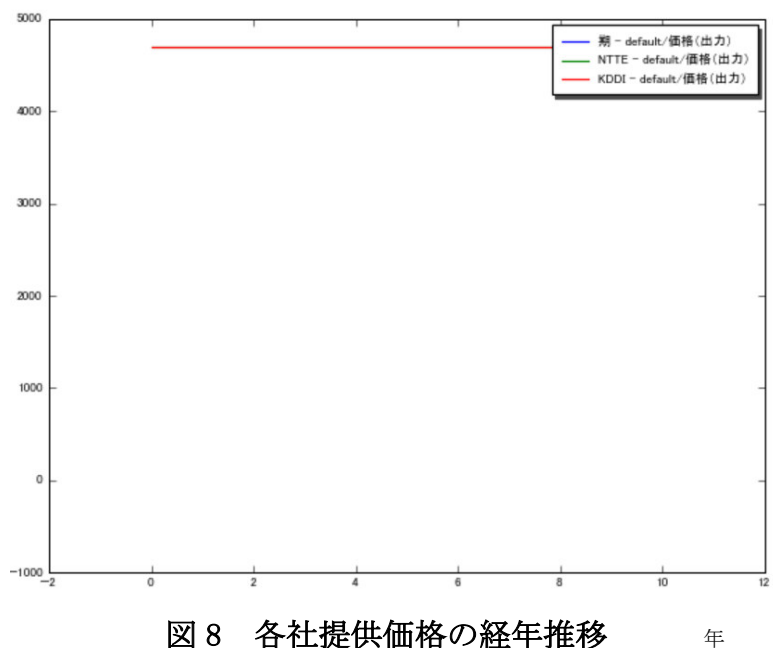


図8 各社提供価格の経年推移

7-1-3. 両モデルの比較

7-1-3-1. NTT ユーザ数推移の比較

NTT 東及びNTT 東卸売業者エージェントの持つNTT ユーザ数を比較すると新モデルの方がユーザを獲得できていることがわかる。

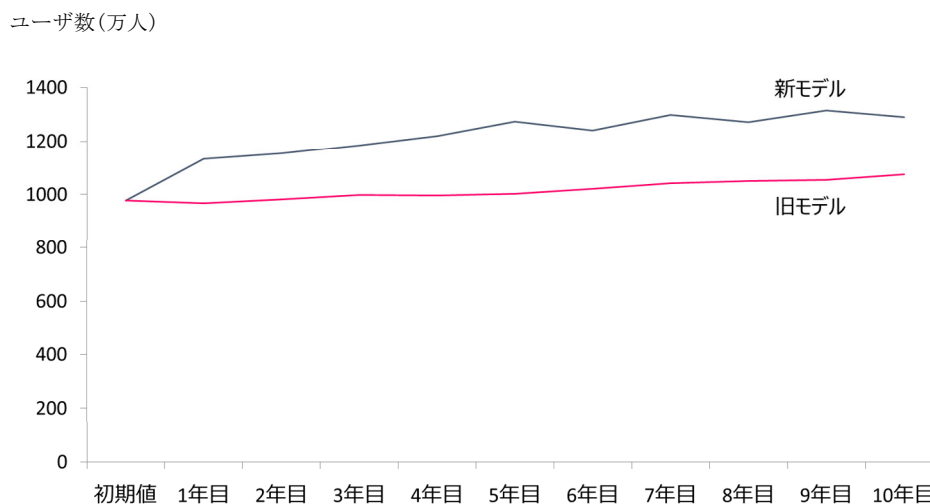


図9 NTT ユーザ数の経年推移

7-1-3-2. 年別 NTT 収益性の比較

i) 年別 NTT 収益性

図10に年別のNTT収益性を示す。NTT収益性を比較すると、旧モデルの方が収益性が高いことがわかる。

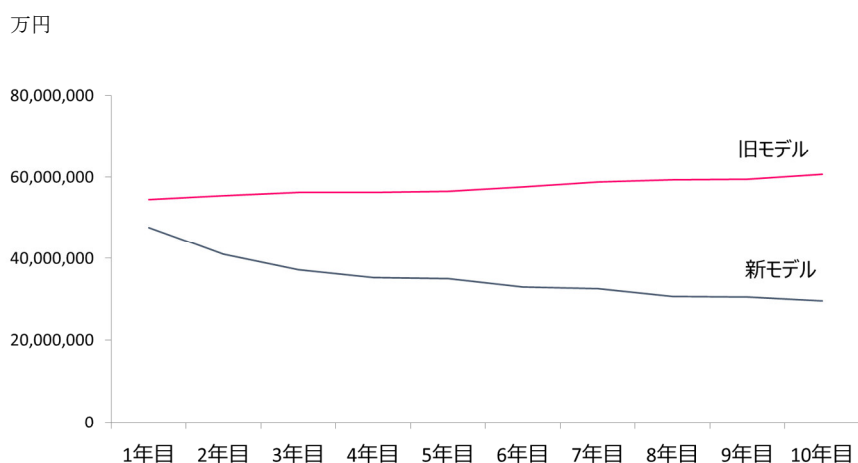


図10 年別 NTT 収益性

ii) 変動係数

a) 新モデル(B2B2C)

図 11 と表 1 に新モデルのシミュレーションを 10 回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

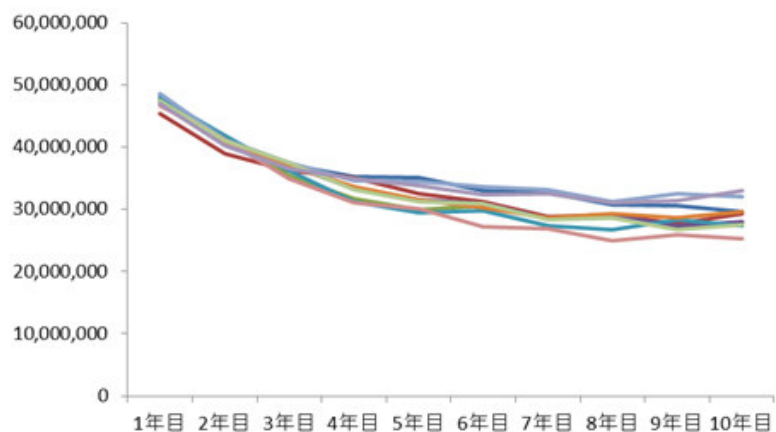


図 11 新モデルの繰り返しシミュレーション結果

表 1 新モデル (平均値、標準偏差、変動係数)

	平均値	標準偏差	変動係数
1年目	47,355,906	887,014	2%
2年目	40,807,584	800,467	2%
3年目	36,567,204	822,699	2%
4年目	33,417,642	1,510,789	5%
5年目	31,959,534	1,864,607	6%
6年目	30,998,280	1,707,790	6%
7年目	29,554,662	2,166,440	7%
8年目	28,934,646	1,863,212	6%
9年目	28,594,908	2,071,241	7%
10年目	28,953,726	2,177,307	8%

b) 旧モデル(B2C)

図12と表2に旧モデルのシミュレーションを10回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

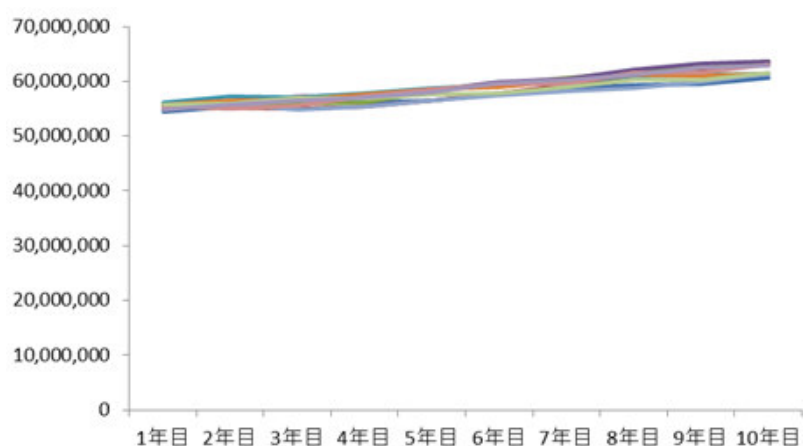


図12 旧モデルの繰り返しシミュレーション結果

表2 旧モデル（平均値、標準偏差、変動係数）

	平均値	標準偏差	変動係数
1年目	55,413,000	485,990	1%
2年目	55,892,400	604,296	1%
3年目	56,253,360	802,071	1%
4年目	56,744,040	824,463	1%
5年目	57,736,680	869,384	2%
6年目	58,734,960	931,949	2%
7年目	59,620,440	775,146	1%
8年目	60,816,120	1,038,756	2%
9年目	61,391,400	1,183,124	2%
10年目	62,254,320	1,071,837	2%

iii) 平均値の差の検定

旧モデルと新モデルの累積収益性の結果について平均値の差の検定を実施。

P 値 : $6.8E^{-10}$

結果より、係数が正で有意(5%水準)のため、旧モデルと新モデルの平均値に有意な差があるということが言える。

7-1-3-3. 累積 NTT 収益性の比較

i) 累積 NTT 収益性

図 11 に累積の NTT 収益性を示す。NTT 収益性を比較すると、旧モデルの方が収益性が高いことがわかる。

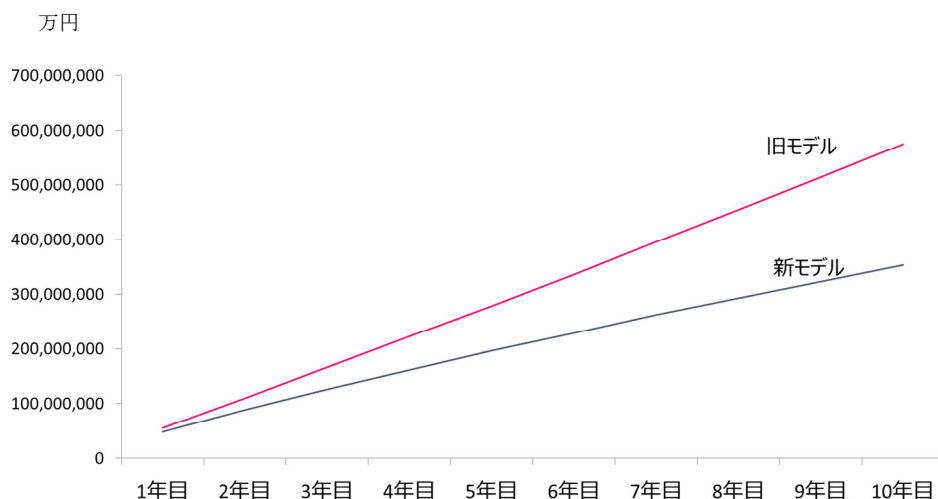


図 13 累積 NTT 収益性

ii) 変動係数

a) 新モデル (B2B2C)

図 14 と表 3 に新モデルのシミュレーションを 10 回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

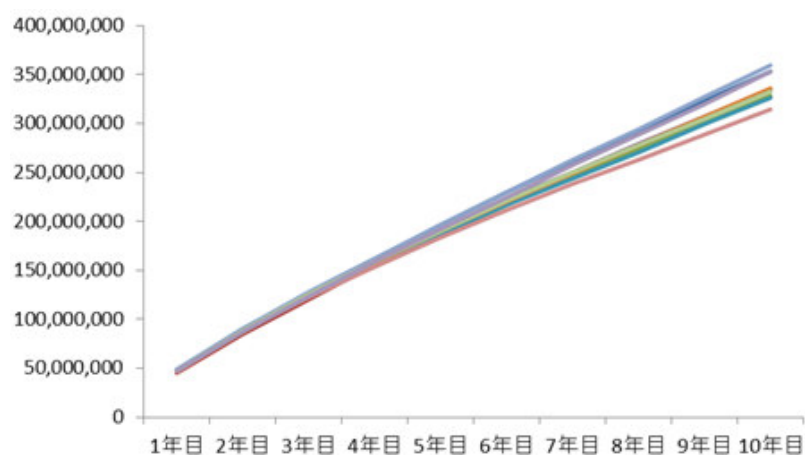


図 14 新モデルの繰り返しシミュレーション結果

表3 新モデル（平均値、標準偏差、変動係数）

	平均値	標準偏差	変動係数
1年目	55,413,000	485,990	1%
2年目	55,892,400	604,296	1%
3年目	56,253,360	802,071	1%
4年目	56,744,040	824,463	1%
5年目	57,736,680	869,384	2%
6年目	58,734,960	931,949	2%
7年目	59,620,440	775,146	1%
8年目	60,816,120	1,038,756	2%
9年目	61,391,400	1,183,124	2%
10年目	62,254,320	1,071,837	2%

a) 旧モデル(B2C)

図15と表4に旧モデルのシミュレーションを10回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

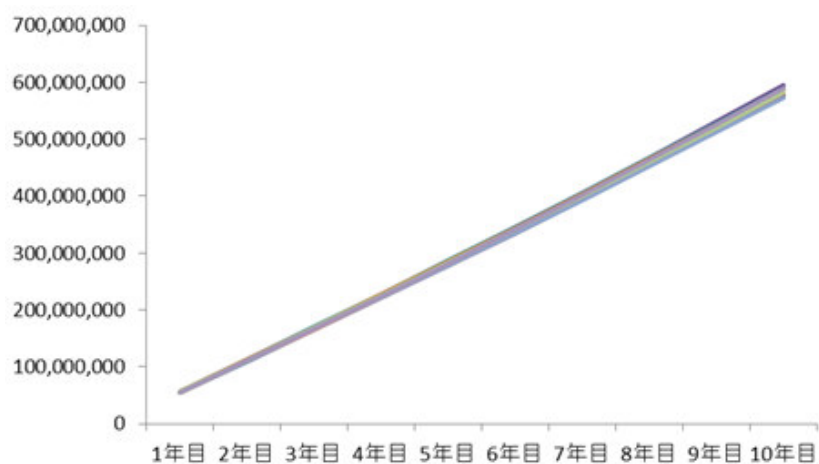


図15 旧モデルの繰り返しシミュレーション結果

表4 旧モデル（平均値、標準偏差、変動係数）

	平均値	標準偏差	変動係数
1年目	55,413,000	485,990	1%
2年目	111,305,400	1,024,634	1%
3年目	167,558,760	1,644,802	1%
4年目	224,302,800	2,357,898	1%
5年目	282,039,480	3,086,173	1%
6年目	340,774,440	3,759,342	1%
7年目	400,394,880	4,339,195	1%
8年目	461,211,000	5,159,737	1%
9年目	522,602,400	5,984,356	1%
10年目	584,856,720	6,569,174	1%

iii) 平均値の差の検定

旧モデルと新モデルの結果について平均値の差の検定を実施。

P 値： $1.2E^{-5}$

結果より、係数が正で有意(5%水準)のため、旧モデルと新モデルの平均値に有意な差があるということが言える。

7-2. 検証②:顧客の乗換え周期が変わった場合の検証

7-2-1. 顧客の乗換え周期が長くなったシナリオ

7-2-1-1. 新モデル(B2B2C)

図16に各社のユーザ数の経年推移を示す。NTT東エージェントがユーザ数を減らす一方、KDDIエージェント及びNTT東卸売業者エージェントがユーザ数を伸ばしていることが読み取れる。

ユーザ数(万人)

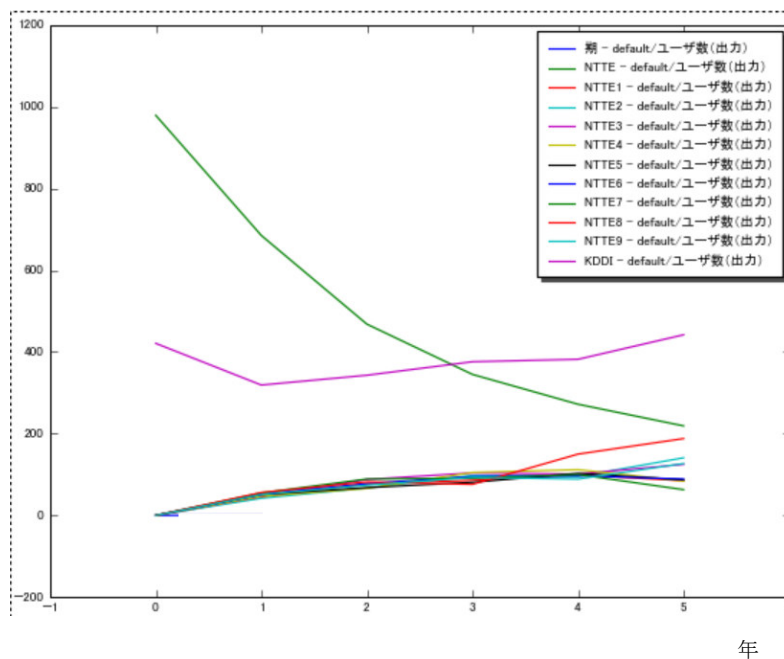


図 16 各社ユーザー数の経年推移

また、図 17 に各社の提供価格の経年推移を示す。NTT 東エージェント以外は価格競争を行い価格を年々下げていることがわかる。

月額提供価格(円)

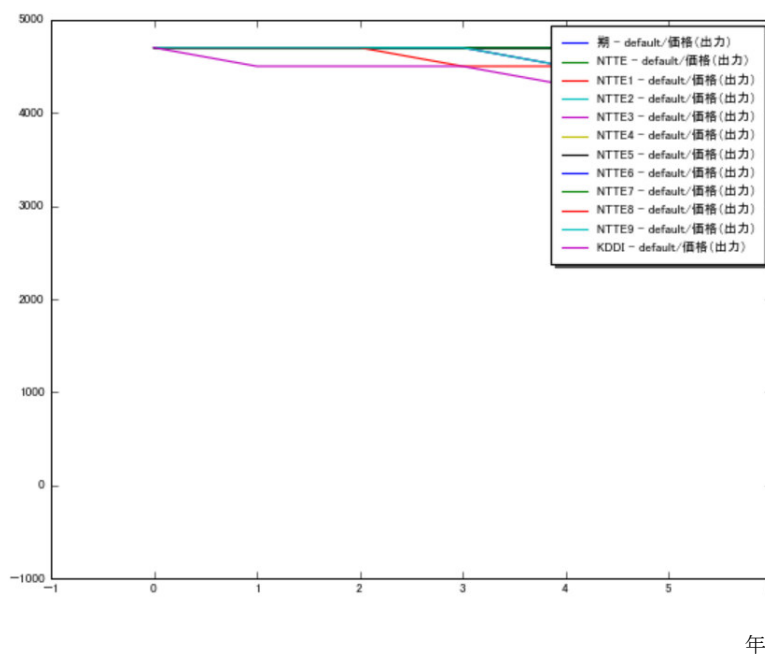


図 17 各社提供価格の経年推移

7-2-1-2. 旧モデル(B2C)

図 18 に各社のユーザ数の経年推移を示す。NTT 東エージェント、KDDI エージェントともに伸びているが KDDI エージェントの伸びの方が大きいことがわかる。

ユーザ数 (万人)

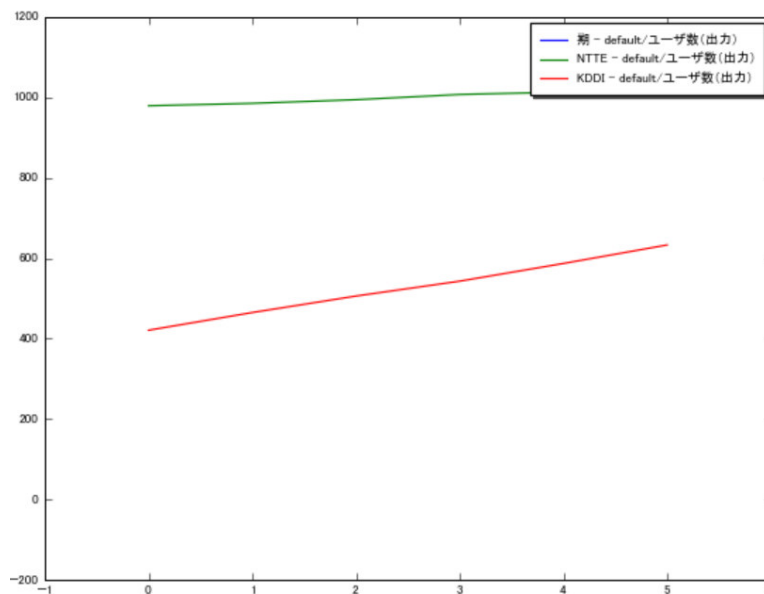


図 18 各社ユーザ数の経年推移 年

また、図 19 より、価格競争を行っていないことから、NTT 東、KDDI ともに価格は一定であることがわかる。

月額提供価格(円)

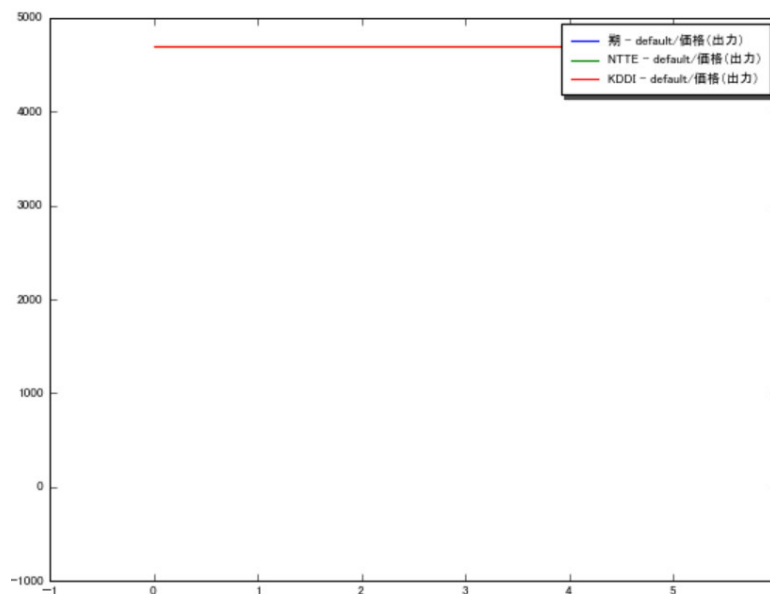


図 19 各社提供価格の経年推移 年

7-2-1-3. NTT ユーザ数推移の比較

図 20 に NTT ユーザ数の経年推移を示す。旧モデルは微減する一方、新モデルは増加しており、新モデルの方がユーザを確保できていることが読み取れる。

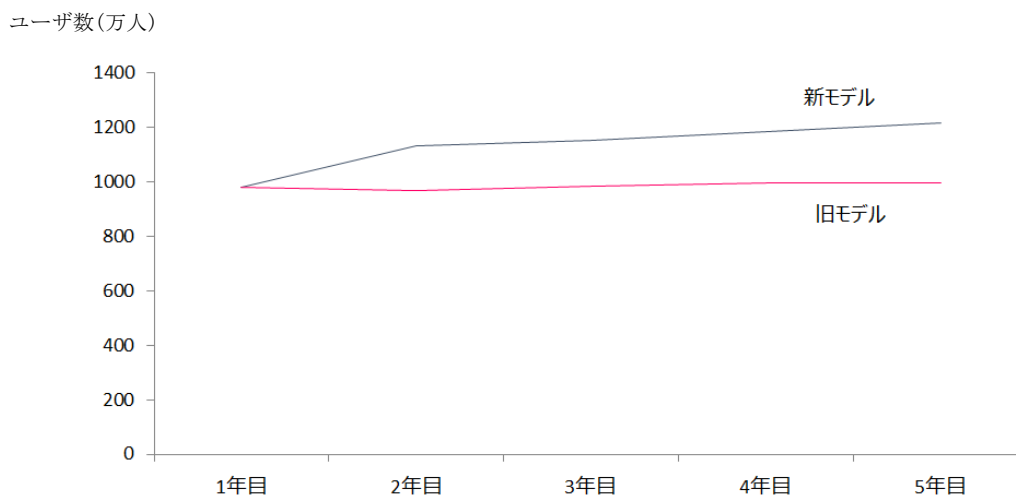


図 20 NTT ユーザ数の経年推移

7-2-1-4. 年別 NTT 収益性の比較

i) 年別 NTT 収益性

図 21 に年別の NTT 収益性を示す。旧モデルは収益性を維持している一方、新モデルは収益性を落としていることがわかる。

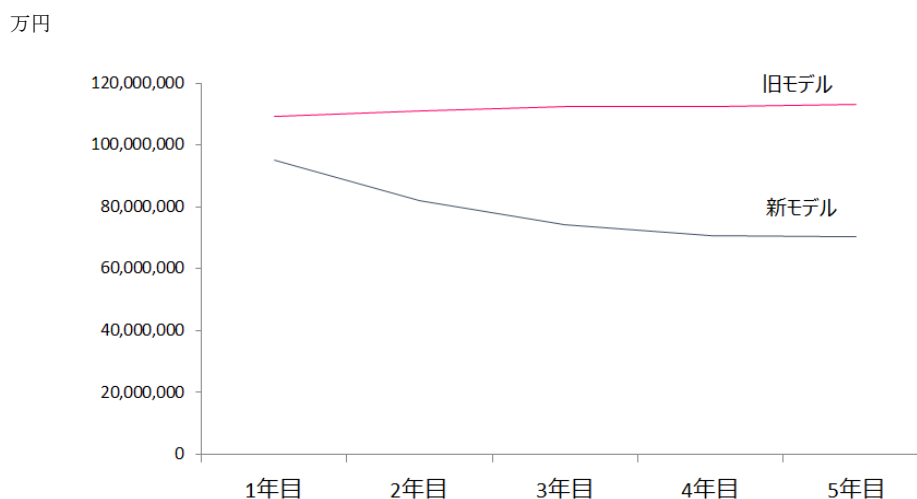


図 21 年別 NTT 収益性

ii) 変動係数

a) 新モデル(B2B2C)

図 22 と表 5 に新モデルのシミュレーションを 10 回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

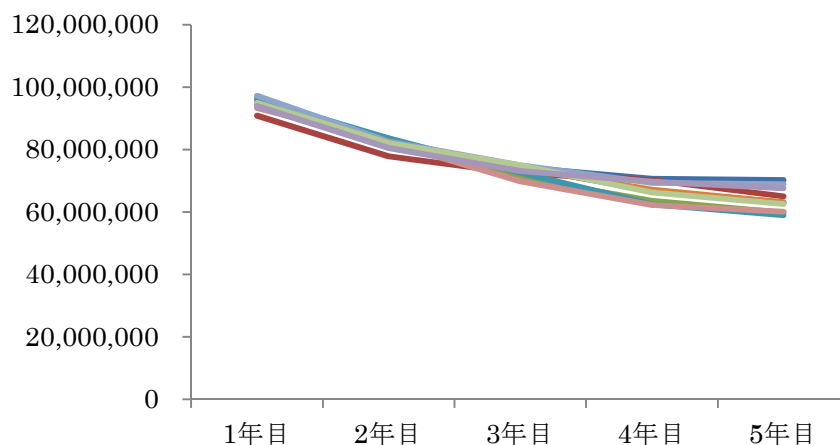


図 22 新モデルの繰り返しシミュレーション結果

表 5 新モデル (平均値、標準偏差、変動係数)

	平均	標準偏差	変動係数
1年目	94,711,812	1,774,028	2%
2年目	81,615,168	1,600,933	2%
3年目	73,134,408	1,645,398	2%
4年目	66,835,284	3,021,578	5%
5年目	63,919,068	3,729,214	6%

b) 旧モデル(B2C)

図 23 と表 6 に旧モデルのシミュレーションを 10 回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

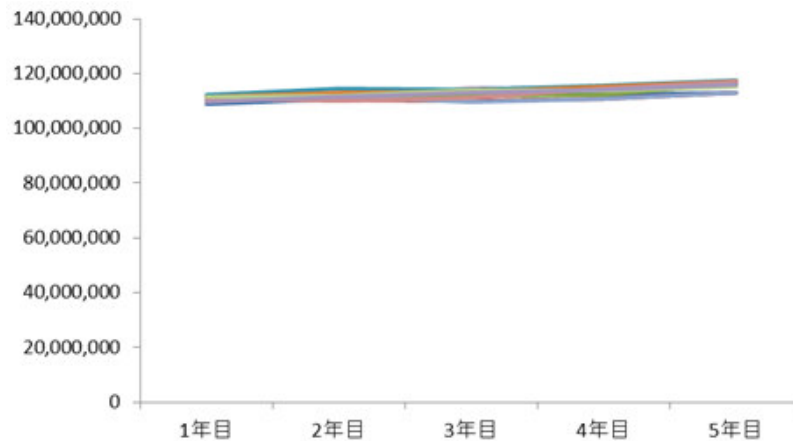


図 23 旧モデルの繰り返しシミュレーション結果

表 6 旧モデル（平均値、標準偏差、変動係数）

	平均	標準偏差	変動係数
1年目	110,826,000	971,980	1%
2年目	111,784,800	1,208,593	1%
3年目	112,506,720	1,604,141	1%
4年目	113,488,080	1,648,926	1%
5年目	115,473,360	1,738,767	2%

iii) 平均値の差の検定

旧モデルと新モデルの結果について平均値の差の検定を実施。

P 値： $1.6E^{-9}$

結果より、係数が正で有意(5%水準)のため、旧モデルと新モデルの平均値に有意な差があるということが言える。

7-2-1-5. 累積 NTT 収益性の比較

i) 累積 NTT 収益性

図 24 に累積の NTT 収益性を示す。NTT 収益性を比較すると、旧モデルの方が収益性が高いことがわかる。

万円

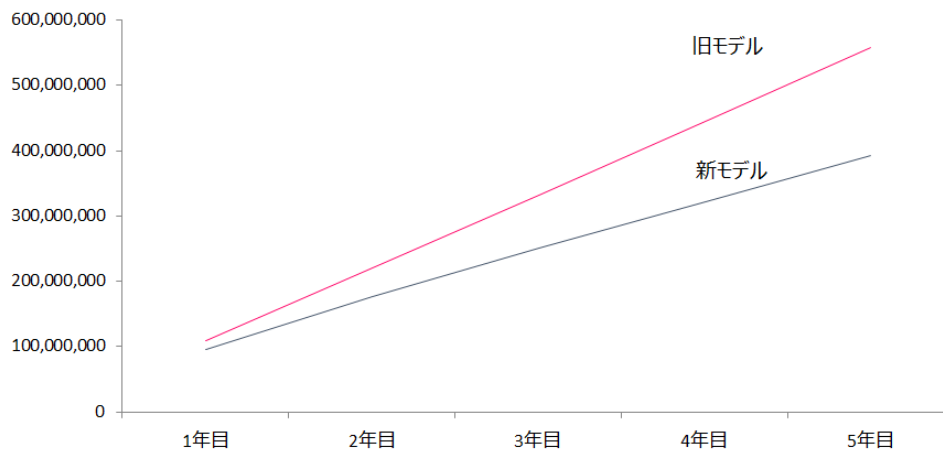


図 24 累積 NTT 収益性

ii) 変動係数

a) 新モデル (B2B2C)

図 25 と表 7 に新モデルのシミュレーションを 10 回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

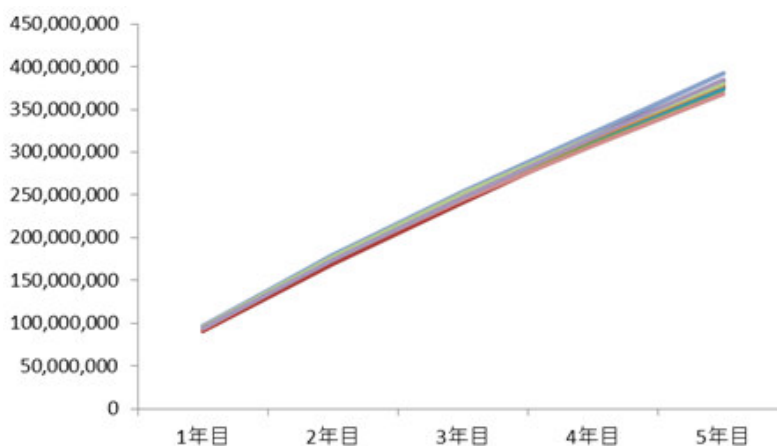


図 25 新モデルの繰り返しシミュレーション結果

表 7 新モデル (平均値、標準偏差、変動係数)

	平均	標準偏差	変動係数
1年目	94,711,812	1,774,028	2%
2年目	176,326,980	3,067,899	2%
3年目	249,461,388	3,808,485	2%
4年目	316,296,672	4,704,367	1%
5年目	380,215,740	7,801,220	2%

b) 旧モデル(B2C)

図 26 と表 8 に旧モデルのシミュレーションを 10 回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

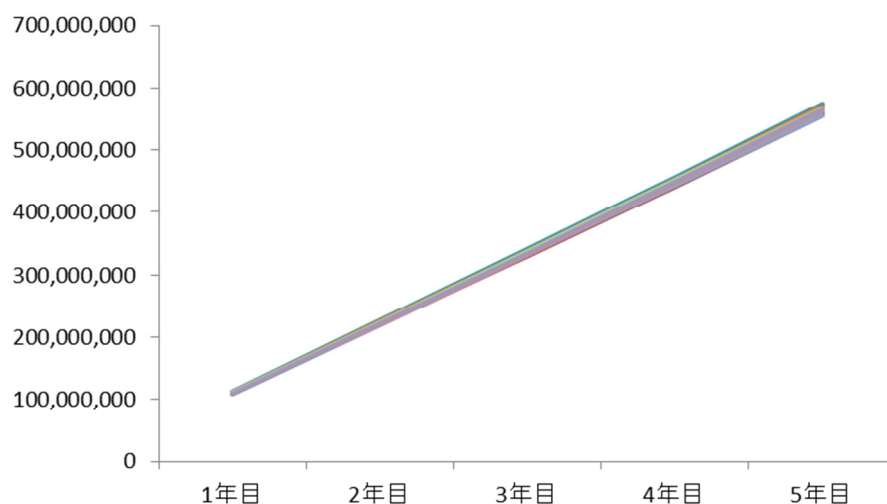


図 26 旧モデルの繰り返しシミュレーション結果

表 8 旧モデル (平均値、標準偏差、変動係数)

	平均	標準偏差	変動係数
1年目	110,826,000	971,980	1%
2年目	222,610,800	2,049,269	1%
3年目	335,117,520	3,289,605	1%
4年目	448,605,600	4,715,796	1%
5年目	564,078,960	6,172,345	1%

iii) 平均値の差の検定

旧モデルと新モデルの結果について平均値の差の検定を実施。

P 値 : $1.8E^{-17}$

結果より、係数が正で有意 (5%水準) のため、旧モデルと新モデルの平均値に有意な差があると言える。

7-2-2. 顧客の乗換周期が短くなったシナリオ

7-2-2-1. 新モデル(B2B2C)

図 27 に各社のユーザ数の経年推移を示す。NTT 東エージェントがユー

ザ数を減らす一方、KDDI エージェント及び NTT 東卸売業者エージェントがユーザ数を伸ばしていることが読み取れる。

ユーザ数 (万人)

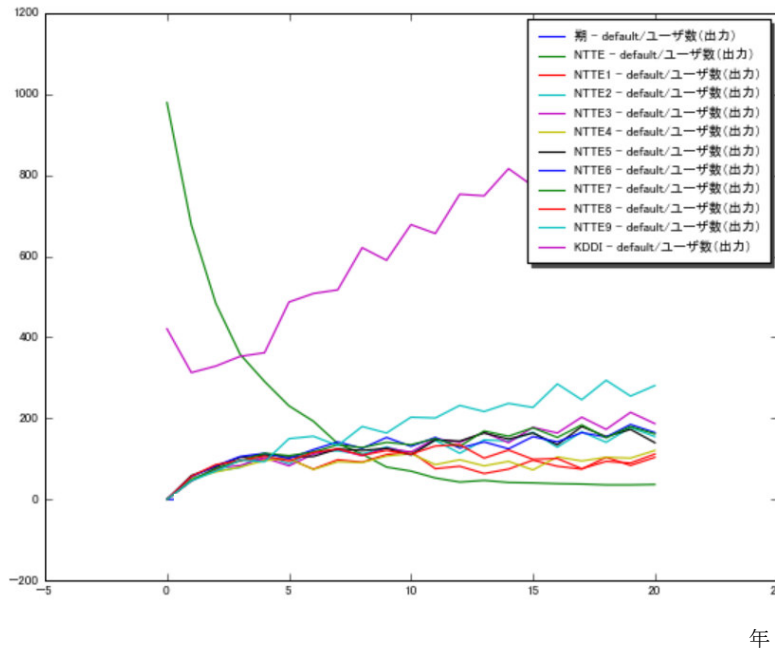


図 27 各社ユーザ数の経年推移

また、図 28 に各社の提供価格の経年推移を示す。NTT 東エージェント以外は価格競争を行い価格を年々下げていることがわかる。

月額提供価格 (円)

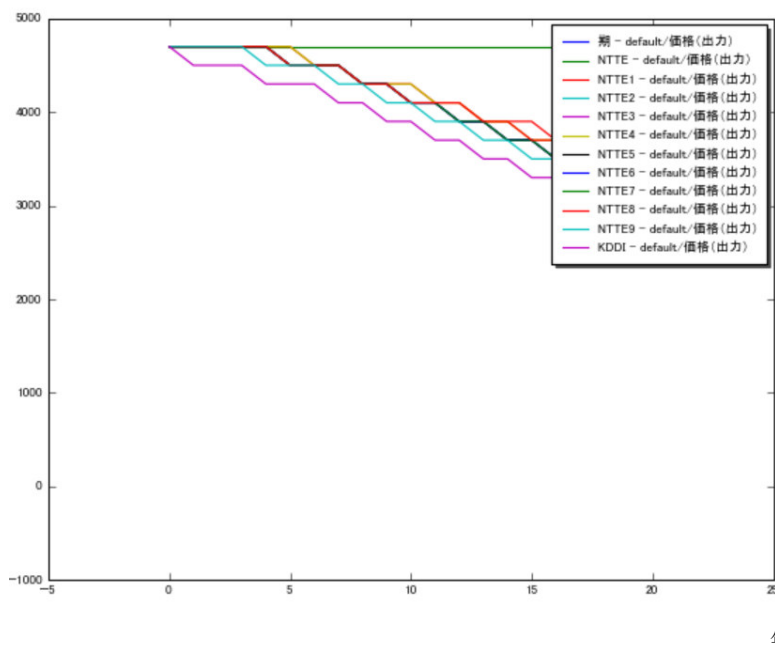


図 28 各社提供価格の経年推移

7-2-2-3. 旧モデル(B2C)

図 29 に各社のユーザ数の経年推移を示す。NTT 東エージェント、KDDI エージェントともに伸びているが KDDI エージェントの伸びの方が大きいことがわかる。

ユーザ数 (万人)

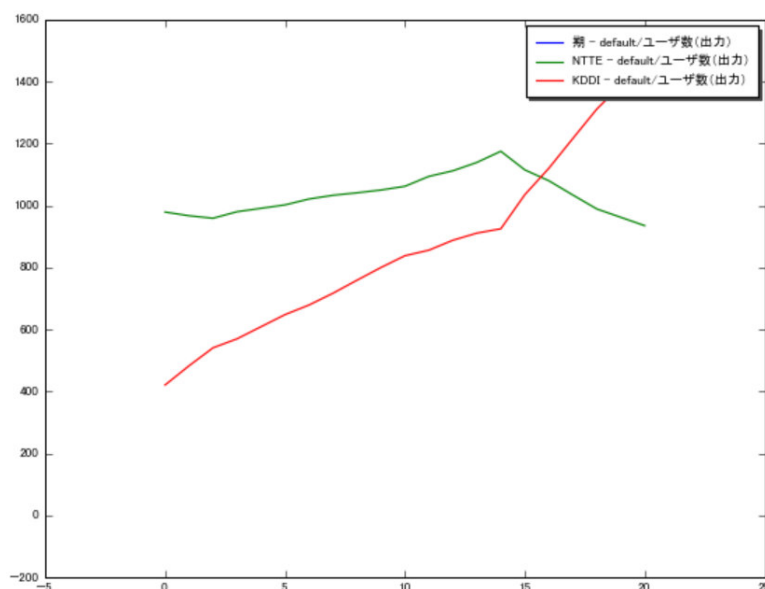


図 29 各社ユーザ数の経年推移 年

また、図 30 より、価格競争を行っていないことから、NTT 東、KDDI ともに価格は一定であることがわかる。

月額提供価格(円)

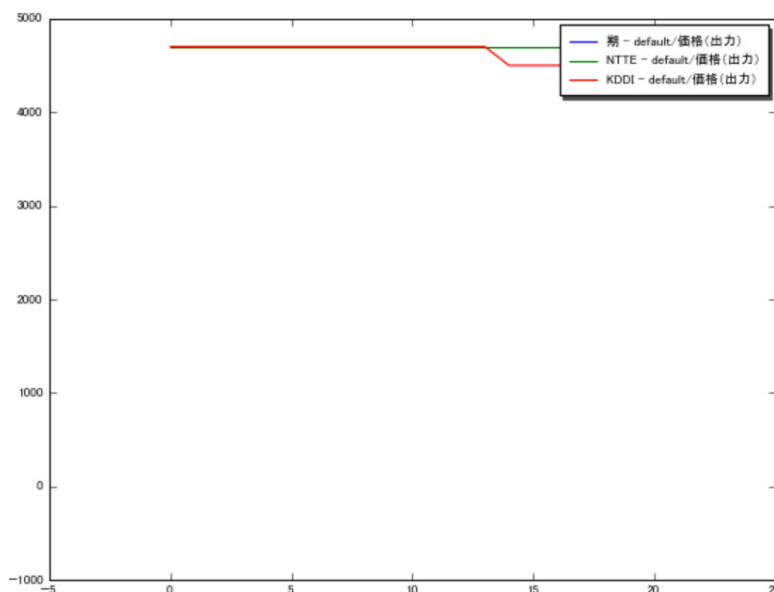


図 30 各社提供価格の経年推移 年

7-2-2-4. NTT ユーザ数推移の比較

図 31 に NTT ユーザ数の経年推移を示す。旧モデルは微減する一方、新モデルは増加しており、新モデルの方がユーザを確保できていることが読み取れる。

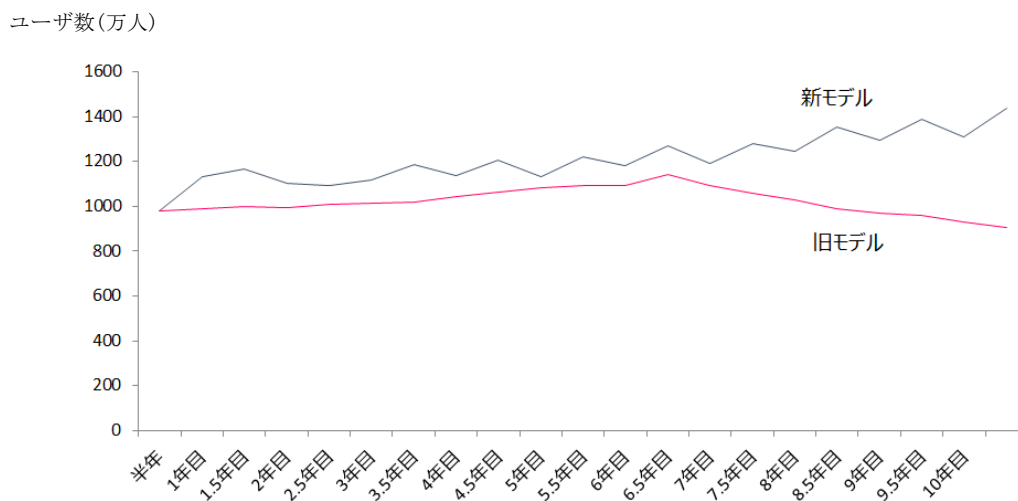


図 31 NTT ユーザ数の経年推移

7-2-2-5. 年別 NTT 収益性の比較

i) 年別 NTT 収益性

図 32 に年別の NTT 収益性を示す。旧モデルは収益性を維持している一方、新モデルは収益性を落としていることがわかる。

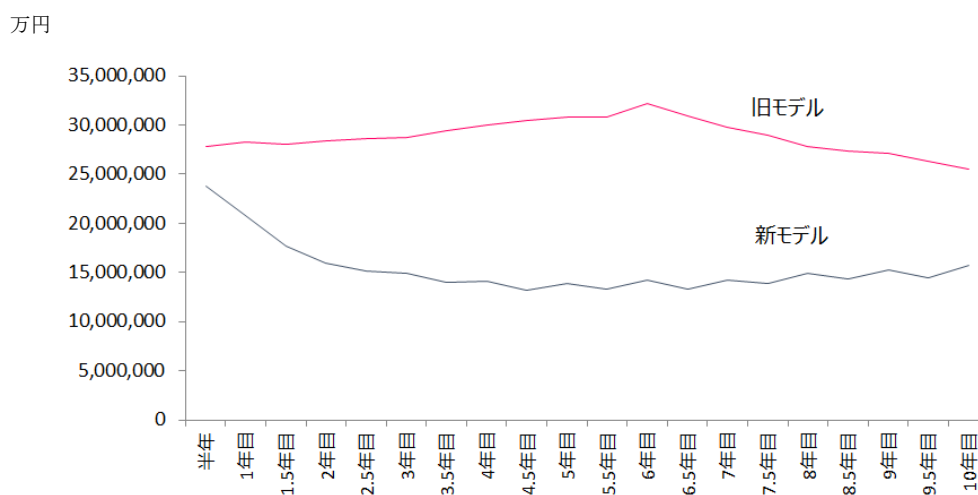


図 32 年別 NTT 収益性

ii) 変動係数

a) 新モデル(B2B2C)

図33と表9に新モデルのシミュレーションを10回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

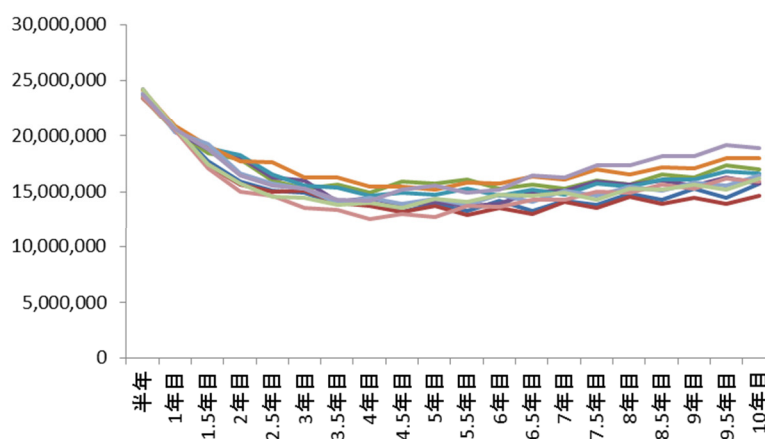


図33 新モデルの繰り返しシミュレーション結果

表9 新モデル（平均値、標準偏差、変動係数）

	平均	標準偏差	変動係数
半年	23,816,457	268,215	1%
1年目	20,596,143	176,611	1%
1.5年目	18,292,608	765,305	4%
2年目	16,741,434	1,116,086	7%
2.5年目	15,707,646	891,702	6%
3年目	15,177,594	706,183	5%
3.5年目	14,476,332	886,045	6%
4年目	14,257,581	726,741	5%
4.5年目	14,213,625	1,009,202	7%
5年目	14,448,006	861,403	6%
5.5年目	14,396,151	1,024,773	7%
6年目	14,580,795	689,647	5%
6.5年目	14,805,741	1,112,039	8%
7年目	15,019,443	704,415	5%
7.5年目	15,319,245	1,214,942	8%
8年目	15,573,177	794,647	5%
8.5年目	15,823,005	1,233,675	8%
9年目	15,970,389	999,731	6%
9.5年目	16,287,270	1,530,883	9%
10年目	16,528,977	1,151,672	7%

b) 旧モデル(B2C)

図 34 と表 10 に旧モデルのシミュレーションを 10 回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

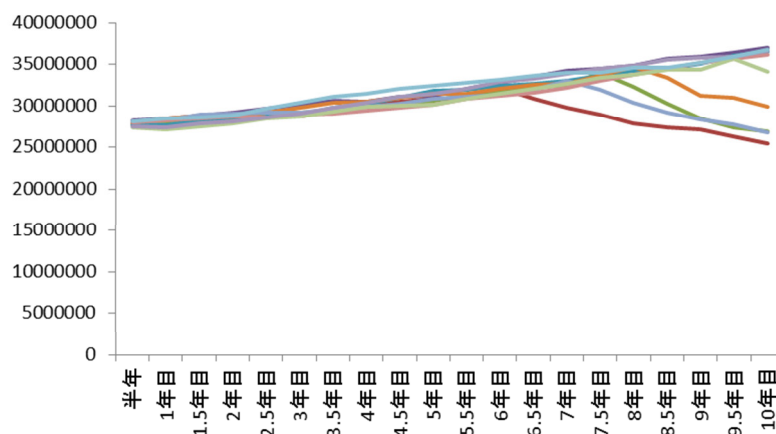


図 34 旧モデルの繰り返しシミュレーション結果

表 10 旧モデル（平均値、標準偏差、変動係数）

	平均	標準偏差	変動係数
半年	27,824,940	266,650	1%
1年目	28,025,160	445,310	2%
1.5年目	28,298,700	486,522	2%
2年目	28,583,520	401,828	1%
2.5年目	28,961,400	418,083	1%
3年目	29,308,260	538,029	2%
3.5年目	29,875,080	610,997	2%
4年目	30,224,760	519,186	2%
4.5年目	30,692,880	634,870	2%
5年目	31,087,680	688,473	2%
5.5年目	31,530,420	631,636	2%
6年目	32,136,720	644,566	2%
6.5年目	32,441,280	831,226	3%
7年目	32,779,680	1,188,837	4%
7.5年目	33,146,280	1,571,284	5%
8年目	33,118,080	2,191,857	7%
8.5年目	32,923,500	2,753,200	8%
9年目	32,666,880	3,332,882	10%
9.5年目	32,807,880	4,012,439	12%
10年目	32,621,760	4,554,729	14%

iii) 平均値の差の検定

旧モデルと新モデルの結果について平均値の差の検定を実施。

P 値： $4.6E^{-23}$

結果より、係数が正で有意(5%水準)のため、旧モデルと新モデルの平均値に有意な差があると言える。

7-2-2-6. 累積 NTT 収益性の比較

i) 累積 NTT 収益性

図 35 に累積の NTT 収益性を示す。NTT 収益性を比較すると、旧モデルの方が収益性が高いことがわかる。

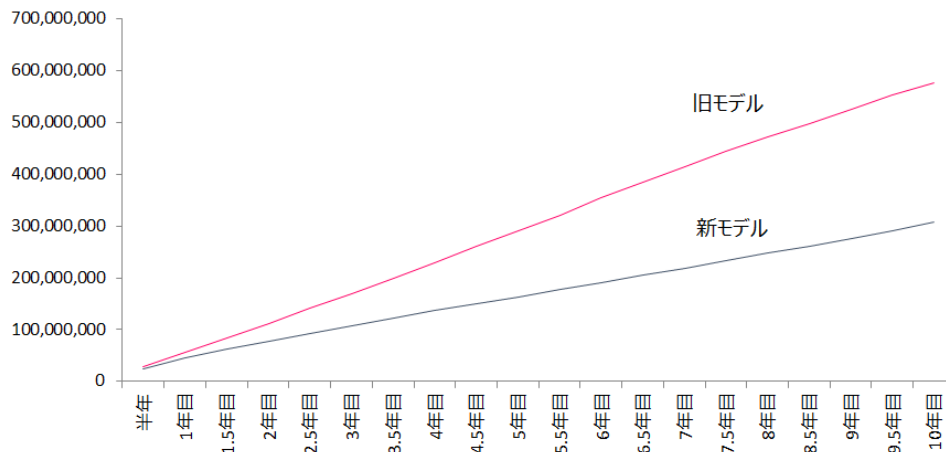


図 35 累積 NTT 収益性

ii) 変動係数

a) 新モデル(B2B2C)

図 36 と表 11 に新モデルのシミュレーションを 10 回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

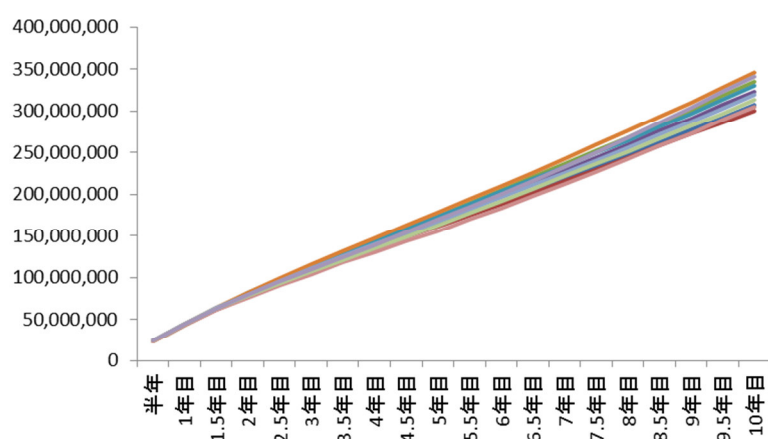


図 36 新モデルの繰り返しシミュレーション結果

表 11 新モデル（平均値、標準偏差、変動係数）

	平均	標準偏差	変動係数
半年	23,816,457	268,215	1%
1年目	44,412,600	331,606	1%
1.5年目	62,705,208	888,242	1%
2年目	79,446,642	1,871,992	2%
2.5年目	95,154,288	2,700,414	3%
3年目	110,331,882	3,351,267	3%
3.5年目	124,808,214	4,087,499	3%
4年目	139,065,795	4,758,880	3%
4.5年目	153,279,420	5,566,492	4%
5年目	167,727,426	6,267,122	4%
5.5年目	182,123,577	7,109,562	4%
6年目	196,704,372	7,685,881	4%
6.5年目	211,510,113	8,571,215	4%
7年目	226,529,556	9,139,802	4%
7.5年目	241,848,801	10,148,111	4%
8年目	257,421,978	10,774,831	4%
8.5年目	273,244,983	11,808,057	4%
9年目	289,215,372	12,655,162	4%
9.5年目	305,502,642	14,002,777	5%
10年目	322,031,619	15,004,018	5%

b) 旧モデル(B2C)

図 37 と表 12 に旧モデルのシミュレーションを 10 回実施した際の結果を示す。平均値、標準偏差、変動係数より、繰り返し実施しても結果に関し大きなばらつきがないことが確認できる。

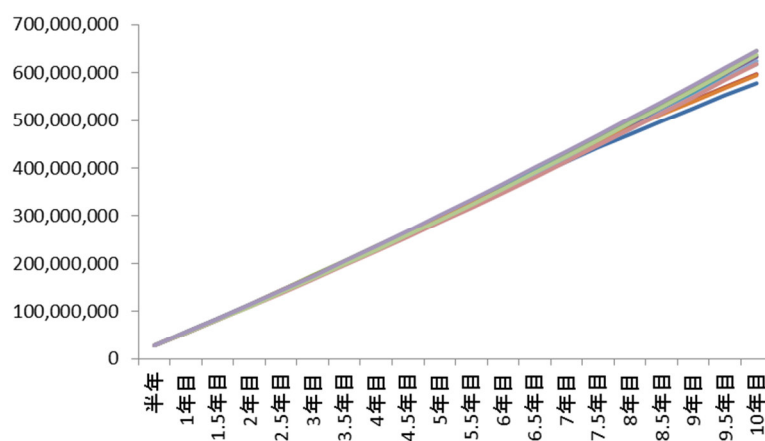


図 37 旧モデルの繰り返しシミュレーション結果

表 12 旧モデル (平均値、標準偏差、変動係数)

	平均	標準偏差	変動係数
半年	27,824,940	266,650	1%
1年目	28,025,160	445,310	2%
1.5年目	28,298,700	486,522	2%
2年目	28,583,520	401,828	1%
2.5年目	28,961,400	418,083	1%
3年目	29,308,260	538,029	2%
3.5年目	29,875,080	610,997	2%
4年目	30,224,760	519,186	2%
4.5年目	30,692,880	634,870	2%
5年目	31,087,680	688,473	2%
5.5年目	31,530,420	631,636	2%
6年目	32,136,720	644,566	2%
6.5年目	32,441,280	831,226	3%
7年目	32,779,680	1,188,837	4%
7.5年目	33,146,280	1,571,284	5%
8年目	33,118,080	2,191,857	7%
8.5年目	32,923,500	2,753,200	8%
9年目	32,666,880	3,332,882	10%
9.5年目	32,807,880	4,012,439	12%
10年目	32,621,760	4,554,729	14%

iii) 平均値の差の検定

旧モデルと新モデルの結果について平均値の差の検定を実施。

P 値： $2.0E^{-8}$

結果より、係数が正で有意(5%水準)のため、旧モデルと新モデルの平均値に有意な差があると言える。

7-2-3. シナリオの比較

項番 7-1, 7-2 の結果より、3 シナリオの比較を行う。

7-2-3-1. NTT ユーザ数の比較

新モデルにおける 3 シナリオの NTT ユーザ数推移を図 38 に示す。10 年目を見ると、顧客の乗換周期が短くなり見直す機会が多いほど NTT のユーザ数は多くなることわかる。

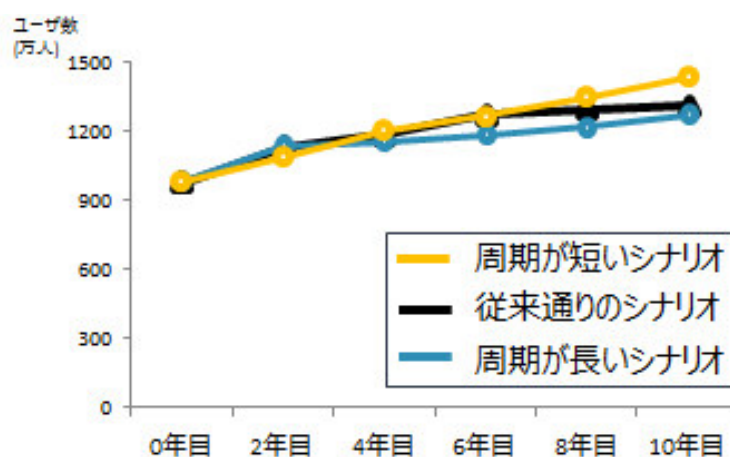


図 38 シナリオ別 NTT ユーザ数推移

7-2-3-2. 累積収益性の比較

新モデルにおける 3 シナリオの累積の収益性を図 39 に示す。10 年目を見ると、顧客の乗換周期が長くなるほど収益性は高くなることわかる。

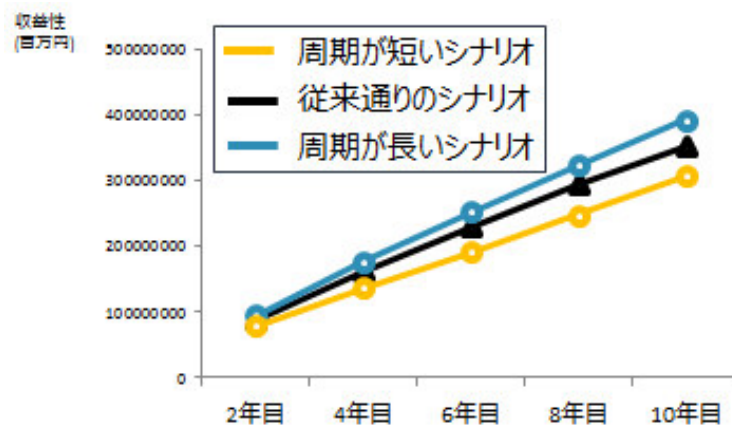


図 39 シナリオ別累積収益性

7-2-3-3. 両モデルの収益差の比較

表 13 に 3 シナリオにおける新旧両モデルの収益差を示す。顧客の乗換周期が短くなるにつれ両モデルの収益差が近づいていくことがわかる。

表 13 各シナリオにおける新モデルと旧モデルの収益差

	顧客乗換周期	新モデル	旧モデル	収益性差
従来通りシナリオ	1年に1回	517百万円	585百万円	-68百万円
周期が長いシナリオ	2年に1回	560百万円	564百万円	-4百万円
周期が短いシナリオ	半年に1回	502百万円	619百万円	-117百万円

7-3. 検証③:市場に新規顧客が増加する場合の検証

項番 7-2-3 より、NTT の新モデルが旧モデルを上回るためには、新規ビジネスによりどの程度の顧客増加を生み出す必要があるかを検証する。表 14 より、顧客の乗換周期が長くなったシナリオの場合、年間 18 万人の増加が見込まれると新シナリオが従来シナリオより収益性が高くなる。通常シナリオに関しては、年間 3.2 百万人の増加が見込まれると収益性が逆転する。一方で、周期が短くなったシナリオでは年間 9.8 百万人の増加が必要となる。2016 年時点での純増数が 40 万人（NTT 東決算資料より）となっているため、今回の環境化ではある程度の市場の増加が必須であると言える。

表 14 各シナリオにおける収益差と必要増加人数

	顧客乗換周期	収益性差	増加人数
従来通りシナリオ	1年に1回	-68百万円	3百万人/年
周期が長いシナリオ	2年に1回	-4百万円	18万人/年
周期が短いシナリオ	半年に1回	-117百万円	5百万人/年

8. 考察

項番 7 の検証結果より、新モデルへの変更により、収益性は下がるもののユーザ獲得数及び市場シェアは上昇することがわかる。更に NTT 東西が直面している外部変化によってその影響度が大きく変わることが検証することができた。まず、1 点目の外部変化である顧客の乗換えの周期が長くなるにつれ、NTT の収益性が上昇する。更にもう 1 点の外部変化である、新規ビジネスの活性化による新規顧客の増加を導くことが出来れば更なる収益性の向上につながる事が証明できた。

今回、私が設定した条件下では旧モデルよりも収益性は低いものの、新規ビジネスを生み出し新規顧客の増加を促進することで旧モデルより収益性をあげることが可能である。また、新規顧客の増加を促進しなくとも、今回想定していない組織の効率化を図ることにより、本意思決定の効果を更に上げることが可能である。

9. 本研究の限界と今後の課題

i) 消費者の効用の定義

消費者の効用は価格のみでは定義できないが、今回は簡易的に価格のみによる影響とした。今後、広告等を含めた価格以外での効用を反映したモデルを作り検証を実施していきたい。

ii) 消費者の効用の経年的な変化を反映できていない

消費者の価格に対する効用も年を経ることに変化が起こるのが現実であるがその点まで考慮することが出来なかった。

iii) 販売店数の増減

現状を見ると年々NTT 東西の卸先業者は増加しており、2016年3月時点では96社(NTT 東日本平成27年度決算資料より)となっている。今回の研究では、販売店数の増加を組み込むことが出来なかった。

10. 参考文献

- 【1】 M. E. ポーター(1995)『競争の戦略』ダイヤモンド社
- 【2】 ゲイリー・ハメル、C・K・プラハラード(2001)『コアコンピタンス経営』日経ビジネス人文庫
- 【3】 山田英夫(1992)『製品ライフサイクルから見た OEM 戦略』「研究・技術・計画」Vol17, No. 3

11. 付属資料

本研究にて使用した S4 のコードの一部を以下に記す。

```
# -*- Mode: Python; coding: utf-8-dos; -*-
```

i) 環境設定

```
from psim import *
from graph import graphclient
import datetime
graphclient.GraphServer.createServer(uri='http://127.0.0.1:63524')

initialize()

setGlobalSeed(None)

# カスタムコード
# なし

# シミュレーションパラメータの定義
class SimulationParam:
    def __init__(self,
                 inputDir = u"input¥¥default",
                 outputDir = u"output¥¥default"):
        self.ncustomers = constantValue(1400).next() # 顧客数
        self.alpha = constantValue(float(10.0 / 3000)).next() # 効用関
        数のスケールパラメータ
        self.epsilon = constantValue(1).next() # この人数の流出が発生し
        たら価格を下げる
        self.delta = constantValue(200).next() # 価格を下げる単位(円)
        self.model = constantValue(1).next() # 1 : モデル 1, 2: モデル 2

# シミュレーターの定義
class Simulator (SimulatorBase):
```

```

def __init__(self,
              param = SimulationParam(),
              inputDir = u"input¥¥default",
              outputDir = u"output¥¥default",
              showGraph = True,
              start = lambda self: None,
              warm = lambda self: 0.0,
              until = lambda self: 5.0,
              odesolver = lambda self: dopri54Solver(dt = 0.1, rtol =
0.001, atol = 1e-06),
              dataDir = u"data"):
    SimulatorBase.__init__(self, param, inputDir, outputDir,
showGraph, start, warm, until, odesolver, dataDir)

# 資源の定義

# オブジェクトの生成
# リンクの生成
# 環境 の定義

class Environment (EnvironmentBase):
    def __init__(self, **keys):
        EnvironmentBase.__init__(self, u"環境", **keys)
    def initAfter(self, **keys):
        u"""環境生成後に呼ばれる。"""
        pass
    def initAttribute(self, agent):
        u"""新規のエージェントが作成されたら、必ずこのメソッド
が呼ばれる。

        エージェントは必ずひとつの環境に属す。

        各エージェントに環境に固有の属性を設定する。"""
        pass
    def setPosition(self, agent, pos):
        u"""エージェント agent の環境に固有の位置属性を更新す

```

る。

エージェントの環境上の位置属性が変わったら、必ずこのメソッドが呼ばれる。

pos は、環境上の位置属性(座標やノード番号など)であり、そのフォーマットは、環境ごとに異なる。"""

```
pass
def getPosition(self, agent):
    u"""エージェント agent の環境に固有の位置属性を返す。
```

環境上の位置属性(座標やノード番号など)であり、そのフォーマットは、環境ごとに異なる。"""

```
pass
def getRandomPosition(self):
    u"""環境内のランダムな位置属性を返す。
```

環境上の位置属性(座標やノード番号など)であり、そのフォーマットは、環境ごとに異なる。"""

```
pass
def draw(self, panel):
    u"""この環境を panel 上に描画する。"""
    pass
def drawAgents(self, panel, agents):
    u"""この環境上で、エージェントのリスト agents を panel
上に描画する。"""
    pass
def custom(self):
    pass
```

```
self.eEnvironment = Environment(simulator = self)
```

```
self.addEnv(self.eEnvironment)
```

ii) 企業エージェントの設定

会社 の定義

```
class CompanySet (SynchronousAgentSetBase):
    def _defAgent(self):
        class _Agent (SynchronousAgentBase):
            def initAfter(self, *args, **keys):

                u"""エージェントが作成され、
environmentBase.initAttribute の呼び出し後に、このメソッドが呼ばれる。
"""

                self.name = keys["name"]
                self.E = keys["E"]
                self.Emin = keys["Emin"]
                self.margin = keys["margin"]
                self.epsilon = keys["epsilon"]
                self.delta = keys["delta"]

            def step(self):
                u"""エージェントのステップ処理"""
                ncurusers = self.count()
                if ncurusers - self.users <= -self.epsilon: # 値
引き判定

                    self.E -= self.delta # 値引き
                    self.E = max(self.E, self.Emin) # 価格の最
低ライン

                    self.users = ncurusers
                def price(self): # 提供価格
                    return self.E + self.margin

                def count(self): # 自社ユーザ数
                    return len([c for c in
self.simulator.aCustomerSet.agents
                                if c.company.name == self.name])

                return _Agent
        def initAfter(self, **keys):
            self.interval = 1.0 # ステップ間隔
```

```

self.freezeAddAgent = True # エージェントの追加を次ステップまで凍結する
self.freezeDelAgent = True # エージェントの削除を次ステップまで凍結する
u"""エージェント集合の作成後にこのメソッドが呼ばれる。

    エージェント集合固有の初期化を行う。"""
self.agentFreezeVars = [""] # エージェントの凍結する変数リスト

# エージェントの生成
self.generateAgents(1,
                    name = "NTTE", E = 4700, Emin = 4700,
                    margin = 0,
                    epsilon = param.epsilon,
                    delta = param.delta)

if param.model == 1:
    for i in xrange(1, 10):
        self.generateAgents(1,
                            name = "NTTE%d" % i,
                            E = 3700, Emin = 1755,
                            margin = 1000,
                            epsilon = param.epsilon,
                            delta = param.delta)

self.generateAgents(1,
                    name = "KDDI", E = 3700, Emin = 1755,
                    margin = 1000,
                    epsilon = param.epsilon,
                    delta = param.delta)

self.usersMonitor = Monitor([u"期"] + [c.name for c in
self.agents],
                             ["i"] * (len(self.agents) +
1),

```

```

name = u"ユーザ数")
self.simulator.addMonitor(self.usersMonitor)
self.pricesMonitor = Monitor([u"期"] + [c.name for c in
self.agents],
name = u"価格")
self.simulator.addMonitor(self.pricesMonitor)
self.earningsMonitor = Monitor([u"期"] + [c.name for c in
self.agents],
name = u"収益")
self.simulator.addMonitor(self.earningsMonitor)

def step(self):
u"""エージェント集合のステップ処理を行う。"""
# エージェント集合へのエージェントの追加削除を凍結
self.freeze()
# 全エージェントを凍結
if len(self.agentFreezeVars) > 0:
for agent in self.agents:
agent.freeze(self.agentFreezeVars)

# 全エージェントのステップ処理
# 会社の前処理
for agent in self.agents:
agent.users = agent.count()

self.usersMonitor.observe(now(), *[c.users for c in
self.agents])
self.pricesMonitor.observe(now(), *[c.price() for c in
self.agents])
self.earningsMonitor.observe(now(), *[c.E - 1755 for c in
self.agents])

```

```

# 顧客のステップ
for customer in self.simulator.aCustomerSet.agents:
    customer.step()
# 会社のステップ
for agent in self.agents:
    agent.step()

# 全エージェントの凍結されていた変更を反映
if len(self.agentFreezeVars) > 0:
    for agent in self.agents:
        agent.melt()
# エージェント集合への凍結されていたエージェントの追加
削除を反映
self.melt()

def start(self):
    u"""同期エージェント集合の動作を開始する。"""
    # エージェント集合のプロセス
    import psim.core
    def proc():
        try:
            while True:
                # ステップ処理
                self.step()
                yield pause(self.interval)
        except psim.core.FinishException, e:
            raise e
        except Exception, e:
            fatal(u "[%s] エージェントエラー: エージェント集
            合のプロセスでエラーが発生しました。" % self.name,
                "error in agentset process", tb = True)
    # エージェント集合のプロセスを起動
    activate(proc)()
def view(self):
    u"""エージェントの描画を開始する。"""
    interval = 1 # 表示間隔

```



```

        #screen = self.getAgentScreen(interval = interval, xlim
= None, ylim = None)
        #screen.addAgentSet(self)
        #screen.start()
    def custom(self):
        pass

    self.aCompanySet = CompanySet(u"会社", self.eEnvironment,
simulator = self)
    self.addAgentSet(self.aCompanySet)

    def startCompanySet():
        self.aCompanySet.start()
        self.aCompanySet.view()

    self.procs.append(startCompanySet)

```

iii) 顧客エージェントの設定

顧客 の定義

```

class CustomerSet (SynchronousAgentSetBase):
    def _defAgent(self):
        class _Agent (SynchronousAgentBase):
            def initAfter(self, *args, **keys):

                u"""エージェントが作成され、
environmentBase.initAttribute の呼び出し後に、このメソッドが呼ばれる。
"""

                self.company = keys["company"]
                self.alpha = keys["alpha"]
            def step(self):
                u"""エージェントのステップ処理"""
                ws = [(numpy.exp(self.alpha *

```

```

self.u(self.company, c)), c)
                for c in
self.simulator.aCompanySet.agents]
                self.company = empiricalDistribution(ws).next()

                def u(self, c0, c1):
                    # c0 から c1 に移行する効用値
                    if c0 == c1:
                        t = 0
                    else:
                        t = -850
                    return - c1.price() + t
                return _Agent
            def initAfter(self, **keys):
                self.interval = 1.0 # ステップ間隔
                self.freezeAddAgent = True # エージェントの追加を次ステ
                ップまで凍結する
                self.freezeDelAgent = True # エージェントの削除を次ステ
                ップまで凍結する
                u"""エージェント集合の作成後にこのメソッドが呼ばれる。

                エージェント集合固有の初期化を行う。"""
                self.agentFreezeVars = [""] # エージェントの凍結する変
                数リスト
                # エージェントの生成 ‘
                n_ntte = int(param.ncustomers * 0.7) # 初期の NTTE のユ
                ーザ数
                n_kddi = param.ncustomers - n_ntte # 初期の KDDI のユ
                ザ数

                companies = self.simulator.aCompanySet.agents

                ntte = [c for c in companies if c.name == "NTTE"][0]
                kddi = [c for c in companies if c.name == "KDDI"][0]

```

```

        self.generateAgents(n_ntte, company = ntte, alpha =
param.alpha)
        self.generateAgents(n_kddi, company = kddi, alpha =
param.alpha)

def step(self):
    u"""エージェント集合のステップ処理を行う。"""
    # エージェント集合へのエージェントの追加削除を凍結
    self.freeze()
    # 全エージェントを凍結
    if len(self.agentFreezeVars) > 0:
        for agent in self.agents:
            agent.freeze(self.agentFreezeVars)
    # 全エージェントのステップ処理
    #for agent in self.agents:
    #    agent.step()

    companies = self.simulator.aCompanySet.agents
    ntte = [c for c in companies if c.name == "NTTE"][0]
    kddi = [c for c in companies if c.name == "KDDI"][0]
    self.generateAgents(35, company = ntte, alpha =
param.alpha)
    self.generateAgents(15, company = kddi, alpha =
param.alpha)

    # 顧客のステップ処理は、会社のエージェント集合から起動
    # している

    # 全エージェントの凍結されていた変更を反映
    if len(self.agentFreezeVars) > 0:
        for agent in self.agents:
            agent.melt()
    # エージェント集合への凍結されていたエージェントの追加
    # 削除を反映
    self.melt()
def start(self):

```

```

u"""同期エージェント集合の動作を開始する。"""
# エージェント集合のプロセス
import psim.core
def proc():
    try:
        while True:
            # ステップ処理
            self.step()
            yield pause(self.interval)
    except psim.core.FinishException, e:
        raise e
    except Exception, e:
        fatal(u "[%s] エージェントエラー: エージェント集
合のプロセスでエラーが発生しました。" % self.name,
            "error in agentset process", tb = True)
# エージェント集合のプロセスを起動
activate(proc)()
def view(self):
u"""エージェントの描画を開始する。"""
interval = 1 # 表示間隔
#screen = self.getAgentScreen(interval = interval, xlim
= None, ylim = None)
#screen.addAgentSet(self)
#screen.start()
def custom(self):
    pass

self.aCustomerSet = CustomerSet(u"顧客", self.eEnvironment,
simulator = self)
self.addAgentSet(self.aCustomerSet)

def startCustomerSet():
    self.aCustomerSet.start()
    self.aCustomerSet.view()

self.procs.append(startCustomerSet)

```

```

def run(self, **args):
    # オブジェクトの起動
    pass
    def startProcs():
        for proc in self.procs:
            proc()
        yield alwaysTrue()
    activate(startProcs)()
    # シミュレーション開始
    start(until = self.until, odesolver = self.odesolver, **args)
    # 結果の保存
    self.save()
    self.flush()
def objective(self):
    s1 = Monitor(name = u"収益", basedir =
self.outputDir)[u"NTTE"].sum()
    if self.param.model == 1:
        for i in xrange(1, 10):
            s1 += Monitor(name = u"収益", basedir =
self.outputDir)[u"NTTE%d" % i].sum()
            s2 = Monitor(name = u"収益", basedir =
self.outputDir)[u"KDDI"].sum()
            print s1, s2
            return s1 - s2

if __name__ == "__main__":
    # 実行時の処理
    simulator = Simulator(param = SimulationParam(),
        inputDir = u"input¥¥default",
        outputDir = u"output¥¥default",
        start = lambda self: None,
        warm = lambda self: 0.0,
        until = lambda self: 5.0,
        odesolver = lambda self: dopri54Solver(dt = 0.1, rtol = 0.001, atol
= 1e-06))

```

```
simulator.run()
result = simulator.calcObjective()
print result
simulator.saveResult(result, u"objective")
if simulator.agentApp is not None:
    print u"エージェント画面の終了待ち受け..."
simulator.join()
sys.exit()
```

12. 謝辞

本修士論文を作成するにあたりご指導いただいた、多くの方々にこの場を借りて御礼を申し上げます。まず、経営に関する点から統計に関する知識まで幅広い点でご指導いただいた主査の高橋教授にお礼を申し上げます。また、戦略についてのご教示をいただいた小林教授と流通についてアドバイスをいただいた副査の坂下教授にもお礼を申し上げます。更に本研究内で使用したエージェントベースモデルのソフトウェアに関し、丁寧にご支援いただいた株式会社NTTデータ数理システムの嶋田様、山本様にも1年間大変お世話になりました。本当に有難う御座いました。