

情報共有と製品開発のモデル化 プリウス開発の事例を手がかりとして

稲水伸行*・上野貴弘**

*東京大学経済学部経済学科 学生

**東京大学大学院総合文化研究科 学生

要約：トヨタ自動車の「プリウス」の開発は、エンジニア間の情報共有化により開発効率が上昇した成功例と言われている。そこで、情報共有化が製品開発効率にどのような影響をもたらすのか、MASによるモデルを構築し、シミュレーションを行うことで考察した。その結果、ある程度の情報共有は効率改善をもたらすが、情報共有を徹底化すると逆効果が現れることがインプリケーションとして得られた。

キーワード：問題解決、情報共有、シミュレーション、MAS

1. はじめに

近年、製品開発における情報の共有化の議論が活発となっている。製品開発の情報を共有化することによって、製品開発の各ステージを並行的に進められると同時に早期に問題を解決することが出来る。その結果、大幅に製品開発のリードタイムを削減することが出来るというのである。

本稿では、まず、「情報共有化」を行い製品開発に取り組んだ事例として、トヨタ自動車の「プリウス」開発の事例を取り上げる。「プリウス」の開発においては、「情報共有化」がプロジェクト進行の重要な柱として位置づけられていた。このプロジェクトにおける情報共有は「メーリングリスト」を使った極めてシンプルかつ限定的なものであった。しかし、それ以前の製品開発に比べて驚くべき効率化を実現したのであった。

一見、広範囲な情報共有こそが問題解決を効率にすると考えがちである。しかし、プリウスの事例を見るとそうでもないらしい。そこで、我々は製品開発プロジェクトの組織を念

頭に、情報共有と製品開発効率の関係を考察するための MAS モデルを構築した。

本稿では、まず、トヨタ自動車の「プリウス」開発の事例を紹介する。次に、製品開発組織を念頭に、MAS によるモデルを構築する。最後に、モデルによるシミュレーションの結果を考察する。

2. トヨタ自動車の「プリウス」開発の事例

本章では、情報共有化によって製品開発の効率を高めた事例として、トヨタ自動車の「プリウス」開発の事例を取り上げる。

「プリウス」は世界初の市販ハイブリッド車である。ガソリンエンジンとモーターの 2 つの動力源を持ち、既存のガソリン自動車に比べて、2 倍の燃費向上を達成した。製品開発においては、エンジンと直結した発電機から供給される交流電流を直流電流に変換してバッテリーに供給するインバーターなどこれまでの自動車にはなかった要素技術の開発と、既存の要素技術をハイブリッドという新規の方式に統合する擦り合わせ¹の 2 点を乗り越えねばならなかった。

後者の解決のためには、技術者間の分野を越えた協力が必要である。その協力が実現するには、その解決に必要な技術者のもとに問題が届く必要がある。この問題伝達速度を高めるために、「プリウス」の開発では、メーリングリストというシステムを活用した。このメーリングリストの効果に関して、開発の責任者であった内山田竹志氏は、「一般論で言いますと、通常は、開発の過程でどこかの担当者が何か問題を発見すると、それをその担当部署の上司に伝え、そこから何らかの追加検討・追加調査などの指示が出て、それを色々やってから、やはりうまくいかないということになったとき、初めて、ようやく CE²に報告が来る。こういう経過をたどります。... (中略) ... 『G21』では、朝に問題が起きたとすると、それをすぐに担当者がメーリングリストに書き込みました。だから、チーフも関係部署も、それを読めば、瞬時に問題が発生したことを知ることが出来ました (p-29)」〔家村 1999〕と述べている。また、開発に参加した朝倉吉隆氏は、「試験車にのせた試作バッテリーの状態を一台一台についてつかんでおく必要が出てきたとき、二十台以上のノートパソコンが必要になったので、技術部門内に電子メールを発信した。普通こうした書類を回してもレスポンス

¹ 自動車は部品間の調整が必要な「擦り合わせ製品」と言われており、もともと擦り合わせの余地が大きい製品である。「プリウス」では、既存の部品をこれまで経験の無い新しいシステムに統合するため、擦り合わせの余地が従来の乗用車より大きかった。

² チーフエンジニアの略称。CE は担当する車の開発の統括者である。

はほとんど期待できないのに、予想していた情報システムの管理部署だけでなく、プリウス開発に直接かわりのない部署からもすぐ回答があった。(p-166)〔碓 1999〕と述べている。このように、メーリングリストを活用することで、問題解決のスピードを上げていったのである。

このメーリングリストは、単にチーム内の情報ネットワークにとどめておくのではなく、チームとして必要であり、また身元を確認できたメンバーについては、次々にリストに加え、拡大していった。最大時には、300人に達していたという。

ハイブリッドという新システムの開発が始まったのが1995年2月、販売開始は1997年12月なので、開発期間は約3年である³。Clark & Fujimoto(1991)によれば、1980年代の日本の自動車メーカーがコンポーネント先行開発を始めてから販売に至るまでの期間は約42ヶ月であり、プリウスの開発がいかに迅速であったかがわかる。

このように、「プリウス」開発の事例では、メーリングリストという2003年現在から見れば極めて簡素な方法で情報共有を図っていた。ただ、それは必ずしも徹底した情報共有を意味していたわけではない。一見、徹底した広範囲な情報共有をしたほうが効率的であると思われるがちである。しかし、「プリウス」開発の事例を見るとそうでもなさそうなのである。

この疑問に答えるために、MASを使ってモデルを構築し、シミュレーションを行うことにした。なぜなら、MASを使えば、比較的簡単に製品開発組織と情報共有化をモデル化することが出来るからである。

3. 問題解決モデルの構築

2章で述べた疑問に答えるために、3章ではMASによるモデルの構築を試みる。本稿は、情報共有と製品開発全体の効率の関係について考察する。従って、ここでの考察の対象とならない製品開発の詳細を大胆に省いて、単純なモデルを構築した⁴。

また、「プリウス」の開発プロセスでみられた情報共有をそのままモデル化するものでもない。プリウスの情報共有のポイントは、「広すぎない範囲での共有」であった。このことがもたらす効果についてのインプリケーションを得るために、共有の範囲を操作できるモデルを構築した。

³ 開発期間には、製品のコンセプト形成の期間を含めることが多い。プリウスの場合、21世紀に向けた戦略車であったため、コンセプト形成に通常より長い時間を要した。「プリウス」のチーフエンジニアの内山氏が製品コンセプトを形成する「G21プロジェクト」のリーダーになったのは、1994年1月であり、この時点製品開発の始点とすると、開発リードタイムは約4年である。

⁴ より一般的な製品開発プロセスをKKMASでモデル化した先駆的試みとして、武藤(2002)がある。

3.1. 問題解決モデルのイメージ

まず、「ある製品開発組織があり、それが多くの問題に直面している」という状況を出発点に、モデルを構築していくことを考えよう。ここで言う、製品開発組織とは、各部門（自動車と言えばエンジン部門、車体部門など）を横断する形で結成されているプロジェクト・チームのことである。

各プロジェクトに参加した人はそれぞれ専門的な技能を持った人たちである。彼らは、問題に直面すると、自分の専門知識をフルに活用して試行錯誤を繰り返しながら解決していく。専門の知識があっても、問題を解決するにはそれなりのエネルギーが必要なのである。また、直面する問題は必ずしも自分の持っている専門分野と関連したものであるとは限らない。このときも涙ぐましい努力を重ねて解決してしまうこともあるかもしれない。しかし、情報共有化していてコミュニケーションが取れているのであれば、その筋の専門家に解決を依頼するということが行われるのが普通であろう。先の朝倉氏の話にあるように、試験評価部門で生じた問題は必ずしもその部門で解決できるのではなく、技術部門等による解決が必要なのである。

我々はこのような製品開発組織の日常的な風景を想定して、次項で説明するように、MASによるモデルを構築した。

3.2. 問題解決モデルのルール

前項で述べた仮定のもと、MASによるモデルを構築していく。

問題解決モデルの初期値は、以下にある通りである。

空間の大きさは 56×56 とする。

問題解決をしていく主体であるエージェントの数は 64 とする。各エージェントは解を一つずつ持ち、空間上に均一に分布しているものとする（図1参照）。

各エージェントはその空間上の位置によって4つの部門に分けられる。また、部門横断的なプロジェクトを7つ用意し、そのどれかに参加するものとする（図2参照）。

問題の数は784とし、空間上に均一に分布させる（図3参照）。ただし、どの問題がどのエージェントに割り当てられるかは、ランダムに決定される⁵。

以上のような初期値設定の下、各エージェントは図4にあるようなルールに基づいて、問題を解決していく。つまり、各エージェントは直面する「問題」と各自の持つ「解」を「エ

⁵ つまり、各エージェントは必ずしも自分の持っている解に近い問題が割り当てられるわけではない。

情報共有と製品開発のモデル化 - プリウス開発の事例を手がかりとして -

エネルギー」を投入して結びつけ、問題解決を図っていくのである。そして、自分では解決できそうになければ、情報共有ネットワークを通じて「その筋」の専門エージェントに問題の解決を依頼する。ちなみにこの情報共有ネットワークは、自分がどのプロジェクト・チームや所属する部門によって決定される（次章参照）。

図1：エージェント（=解）の分布

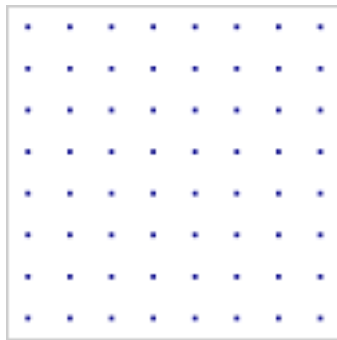
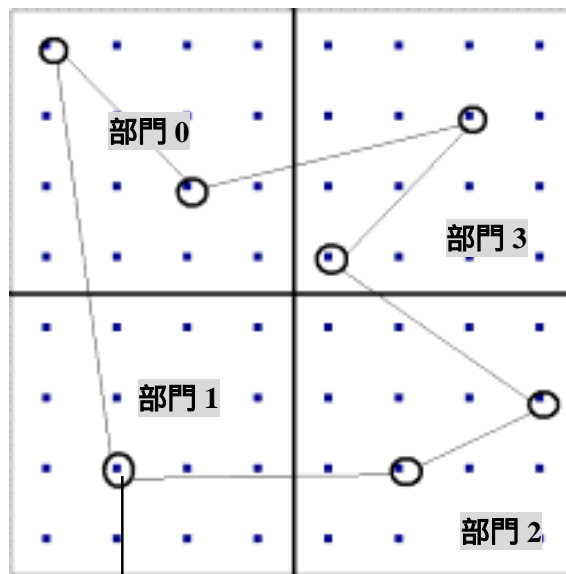


図2：エージェントの部門とプロジェクト・チームの決定



例えば、このエージェントは、部門1に所属することになる。

また、部門横断的なプロジェクト（直線で結んである）に参加している。

図 3 : 問題の分布

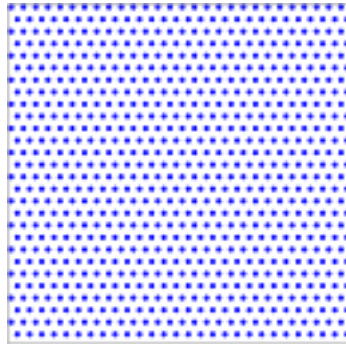
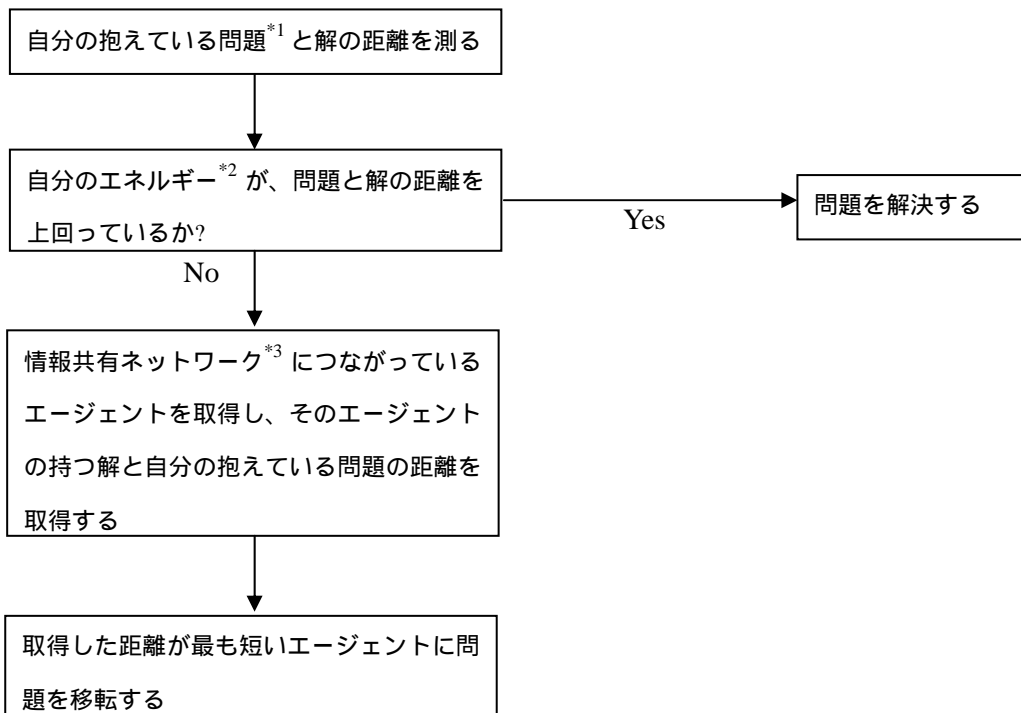


図 4 : エージェントの行動ルール



*1 : 問題は各エージェントにランダムに割り振られる。

*2 : エネルギーは毎ステップ 3 回復する。

*3 : 情報共有ネットワークは次節で説明するように、自らの所属するプロジェクトや部門によって決定される。

4. 問題解決モデルによるシミュレーションとその結果

4.1. シミュレーションのパターンとその結果

3章で説明したモデルにおいて、情報共有ネットワークについて

プロジェクト・チームのみで情報共有をしている場合。

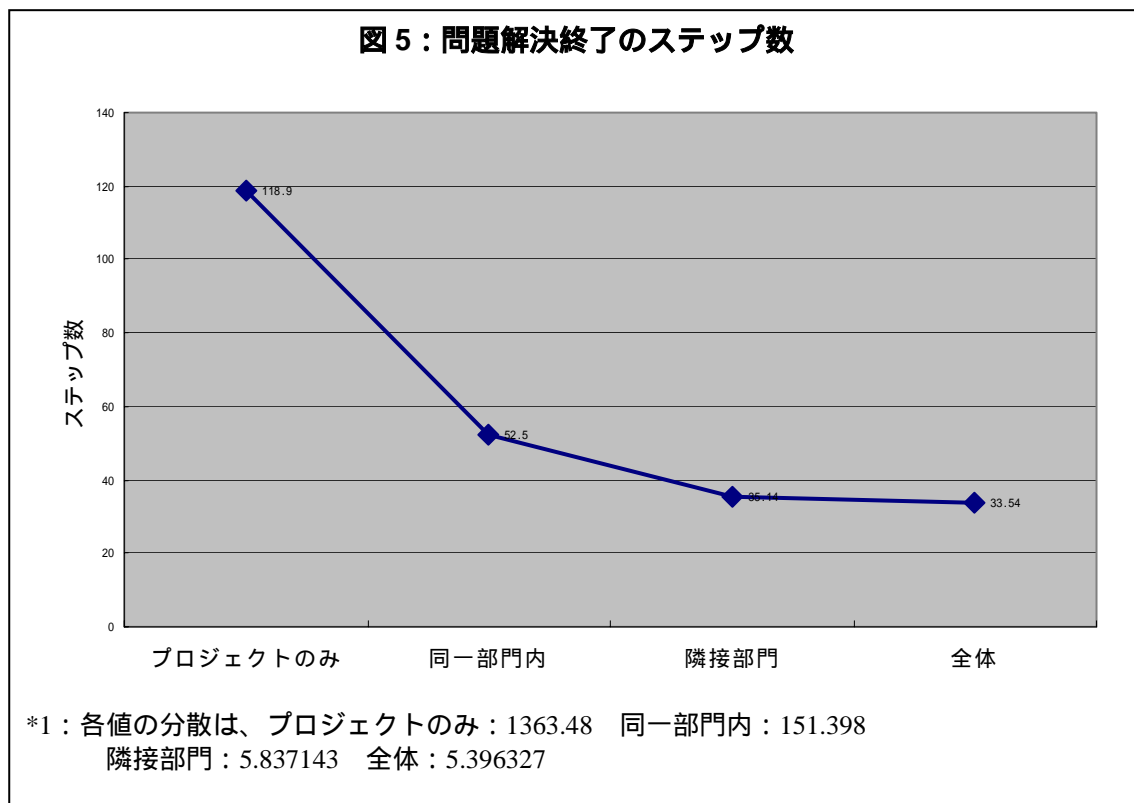
プロジェクト・チームと自分の部門と情報共有をしている場合。

プロジェクト・チームと自分の部門、さらには隣接部門⁶と情報共有をしている場合。

全員と情報共有をしている場合。

の4つの段階についてシミュレーションを行った。それぞれについて、問題が解決されるまでを1回として、50回のシミュレーションを行った。

各場合のシミュレーションにおける問題解決終了ステップ数の平均は図5の通りである。



これを見ると、所属プロジェクト・チーム内のみの情報共有⁷にとどまっている場合には、問題解決効率が非常に悪い。しかし、情報共有の範囲をプロジェクト・チームに加えて所属

⁶ 図2参照。例えば部門1の隣接部門は部門0と部門2になる。

⁷ 以下、便宜上「プロジェクトのみ」と書くことにする。

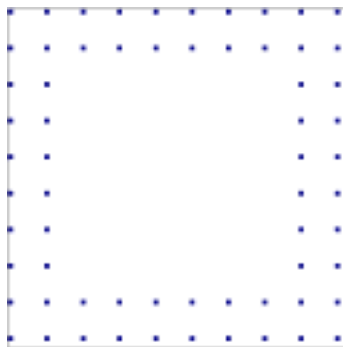
部門内⁸へと拡大するだけで、問題解決効率は著しく上昇する。さらに、隣接部門との情報共有⁹、全体的な情報共有¹⁰へと拡大していくと、問題解決効率は上昇していく。ただ、その上昇率は大きく逡減してしまっている。

4.2. 解 (=エージェント) の分布を変えた場合

以上のシミュレーションでは、解 (=エージェント) の分布も問題の分布も均一で歪みのない場合であった。次に、解 (=エージェント) の分布に歪みがある場合にどのような結果になるのかシミュレーションを行うことにした。

ここで、考えた解 (=エージェント) の分布は ドーナツ型、集中型の二通りである (図6、図7参照)。各場合について、均一な場合 (以下、均一型) と同様に情報共有ネットワークを4段階に変化させて、全ての問題が解決されるまでを1回として50回のシミュレーションを行った。

図6：解 (=エージェント) の分布：ドーナツ型

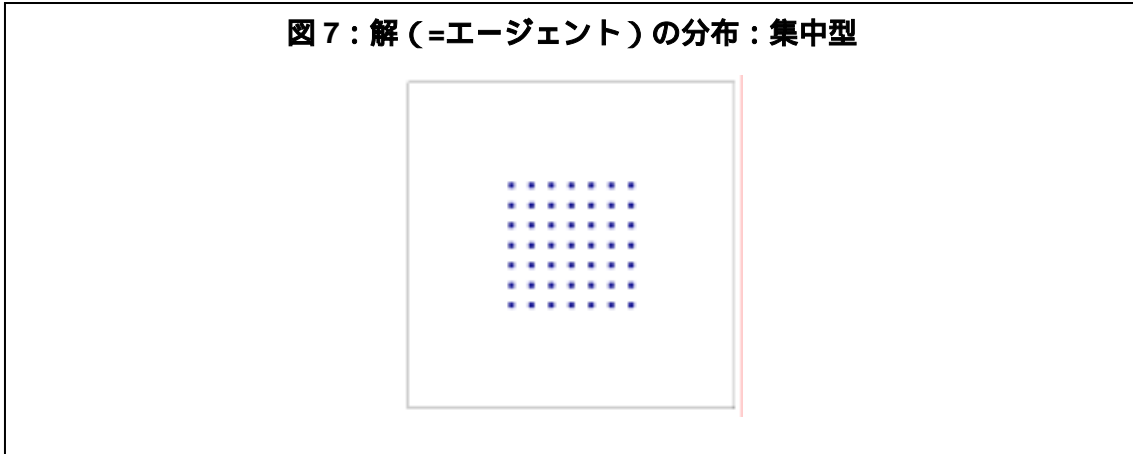


⁸ 以下、便宜上「同一部門内」と書くことにする。

⁹ 以下、便宜上「隣接部門間」と書くことにする。

¹⁰ 以下、便宜上「全体」と書くことにする。

図7：解 (=エージェント) の分布：集中型



このシミュレーションの結果、問題解決終了ステップ数は図8のようになった。まず、ドーナツ型について見ると、均一型に比べて全般的に問題解決終了ステップ数は増えている。また、情報共有の範囲を広げていくと問題解決終了ステップ数は減少していた。しかし、均一な場合に比べてその減少幅は少なくなっている。特に、情報共有の範囲が「プロジェクトのみ」から「同一部門内」へと広げた場合を見ると、均一型では問題解決終了ステップ数が大きく減少したのに対し、ドーナツ型の場合にはあまり減少していない。

次に、集中型について見ていく。この場合、情報共有の範囲を「プロジェクトのみ」にすると最も問題解決終了ステップ数が少ない。そして、情報共有の範囲を「同一部門内」に広げると、かえって問題解決終了ステップ数は大きく増加している。その後、情報共有の範囲を「隣接部門間」、「全体」へと広げてもほとんど変化していない。

解(=エージェント)の分布を変えることで、何が起きたのだろうか。そこで、均一型、ドーナツ型、集中型のそれぞれについて、問題を抱えたエージェント数の推移を見てみた(図9、図10、図11参照)。

図8：解 (=エージェント) の分布を変えた場合の問題解決終了ステップ数

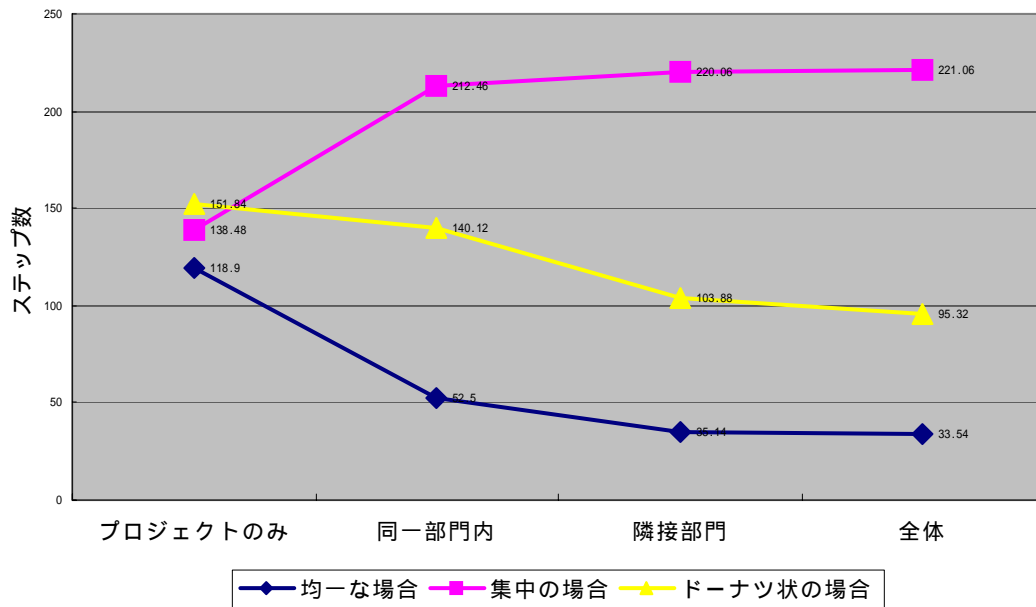


図9：均一型における問題を抱えたエージェント数

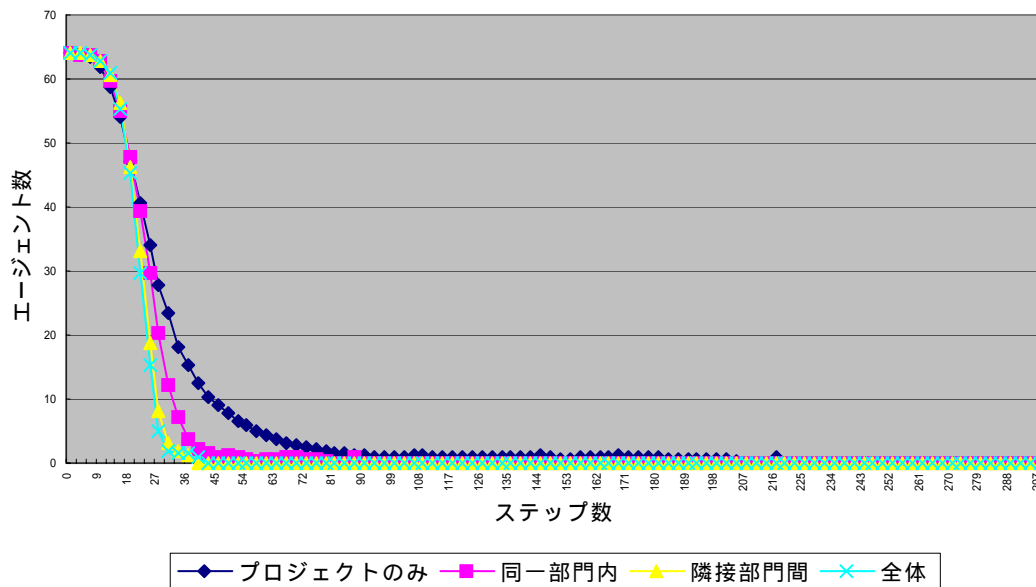


図 10：ドーナツ型における問題を抱えたエージェント数

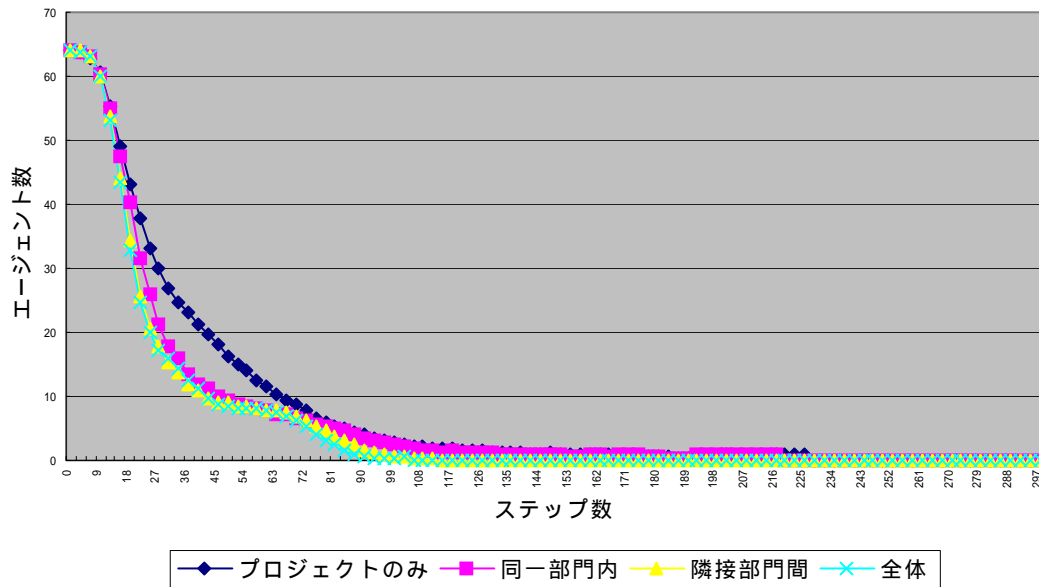


図 11：集中型における問題を抱えたエージェント数

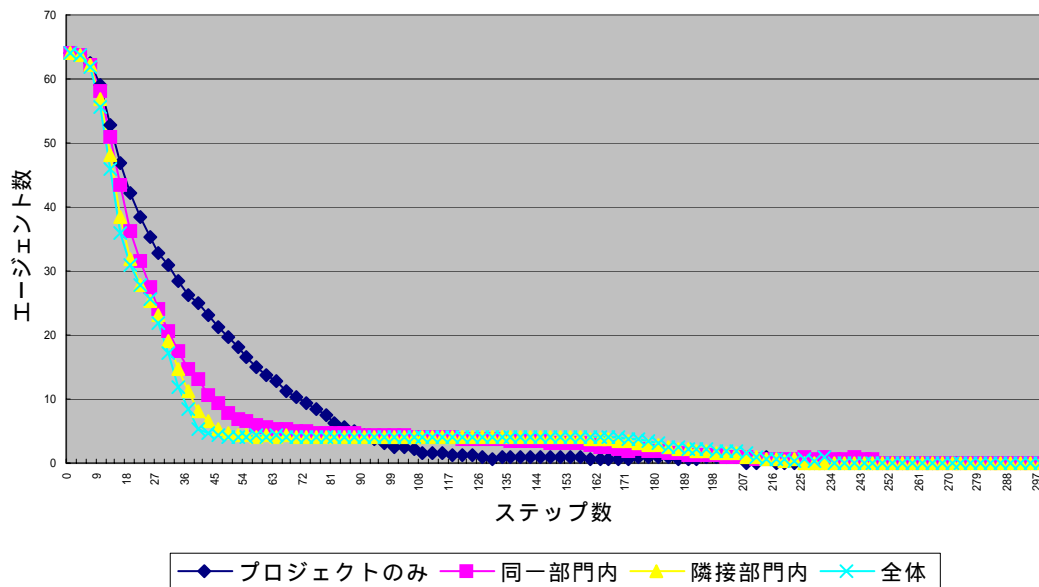


図9を見ると、均一型では、情報共有ネットワークの範囲が「プロジェクトのみ」の場合は他の場合に比べて問題を抱えたエージェント数の減少率があまり大きくない。「プロジェクトのみ」以外の場合は大体同じような減り方をしている。

図10を見ると、ドーナツ型では、「プロジェクトのみ」の場合は、均一型と同様に問題を抱えたエージェント数の減少率はあまり大きくない。それ以外の場合はほとんど同じように推移している。「プロジェクトのみ」以外の3つの場合は、問題を抱えたのエージェント数が10を切る辺りで減少は一度止まり、その後また減少している。

図11を見ると、先の2つの型と同様に、「プロジェクトのみ」に関しては、大きな変化が見られない。その他の3つの場合を見ると、大体残りのエージェント数が5になるあたりで一度減少は止まり、しばらくその状況が続いている。

この状況が一目で分かるのが、図12、図13、図14である。これらは、解 (= エージェント) の分布の各型における「プロジェクトのみ」と「全体」のそれぞれについて、問題を抱えたエージェントを表示したものである。各図では、未解決問題数が300を切った時点と100を切った時点のものを抜き出してきている。

均一型、ドーナツ型、集中型のいずれの場合においても、「全体」の方が「プロジェクトのみ」に比べて問題を抱えているエージェントの数が少ない。問題解決にあたるエージェント数が少ないと、1ステップに問題解決に投入されるエネルギーが少なくなるので、問題解決の効率が悪くなると考えられる。しかし、均一型の「全体」では問題と問題を抱えるエージェントが空間上に一様に分布したまま残っていくので、より問題に近い解を持っているエージェントに問題を移転することによって、「プロジェクトのみ」よりも高い効率を達成することが出来た。

ドーナツ型では、解決が難しい中心部分の問題が残る。情報共有の範囲を「全体」に広げると、これらの問題が真ん中の8つのエージェントに集中する¹¹。この状況では、1ステップに限られたエネルギーしか問題解決に投入されないという非効率が生じる。この結果、問題解決効率が幾分落ちてしまい、8つのエージェントで問題解決に当たるという状況が続くことになる。ただ、ドーナツ型ではかろうじて「プロジェクトのみ」に比べて高い効率を保っている(図8参照)。

しかし、集中型では、情報共有の範囲を「全体」に広げた場合、問題は空間上の四隅に残り、問題を抱えたエージェントもそれに合わせて4つに絞られてしまう。こうなると、1ス

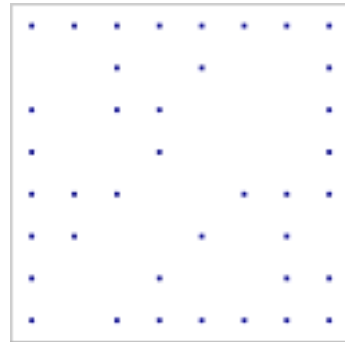
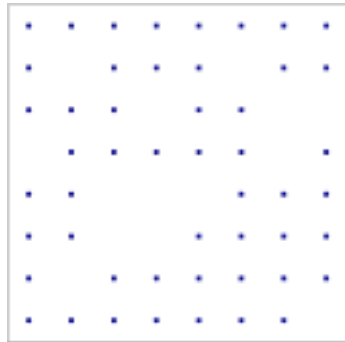
¹¹ この8つのエージェントとは、空間の中心から最も近くなるエージェントである。図13の「全体」の未解決問題数100個時点の図では、この8つのエージェントの他に2つのエージェントも問題を抱えている。

情報共有と製品開発のモデル化 - プリウス開発の事例を手がかりとして -

トップに4つのエージェントが持つエネルギーしか投入されないという大きな非効率が生じる。その結果、「プロジェクトのみ」に比べて、問題解決終了ステップ数が上回るようになってしまったのである。

図 12：問題を抱えたエージェントの推移：均一型

「参加プロジェクトのみ」の未解決問題 300 個時点（左）と 100 個時点（右）



「全体」の未解決問題 300 個時点（左）と 100 個時点（右）

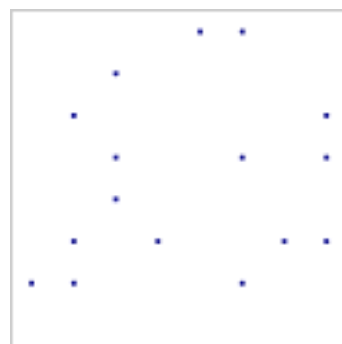
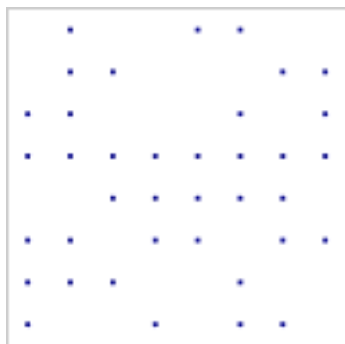
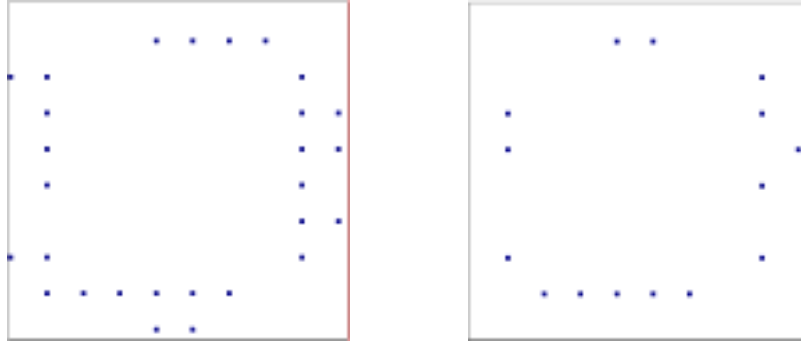


図 13：問題を抱えたエージェントの推移：ドーナツ型

「参加プロジェクトのみ」の未解決問題数 300 個時点（左）と 100 個時点（右）



「全体」の未解決問題数 300 個時点（左）と 100 個時点（右）

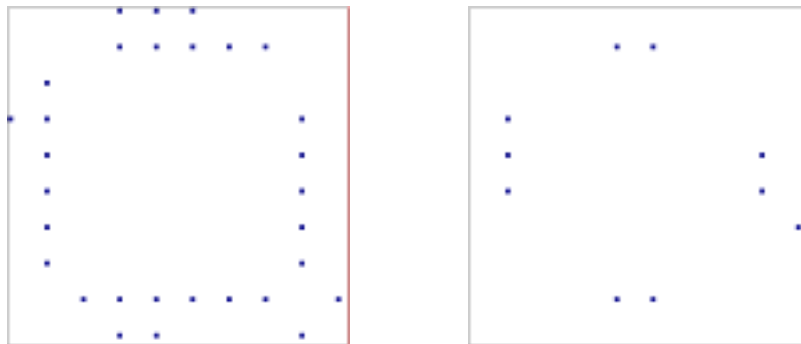
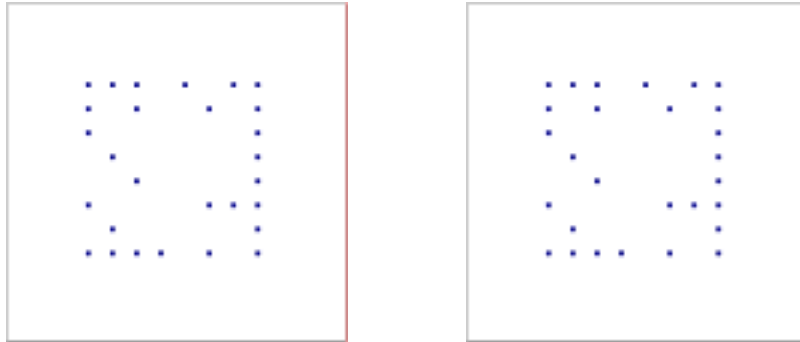
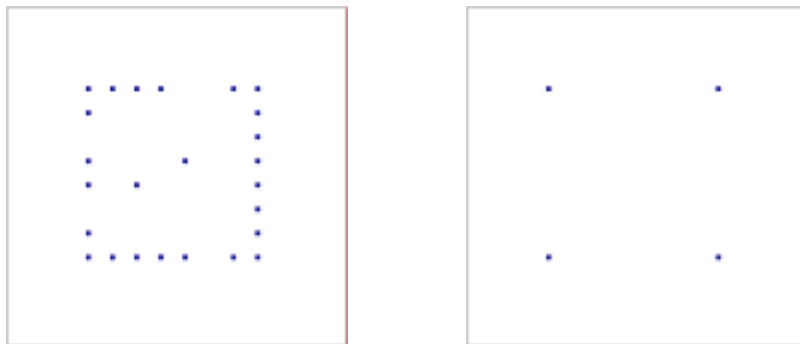


図 14：問題を抱えたエージェント推移：集中型

「参加プロジェクトのみ」の未解決問題数 300 個時点（左）と 100 個時点（右）



「全体」の未解決問題数 300 個時点（左）と 100 個時点（右）



4.3. シミュレーションの結果のまとめ

以上のシミュレーションの結果をまとめると、

一型の場合、情報共有ネットワークの範囲を「プロジェクトのみ」から「部門内」に拡大することで問題解決効率は格段に上昇する。

ただし、情報共有ネットワークの範囲をさらに「隣接部門間」「全体」へと拡大しても、問題解決効率の上昇率は低い。

エージェントの分布に歪みがある場合には、情報共有ネットワークの範囲が広いと一部のエージェントに問題が集中してしまい、結果的に非効率になってしまうことがある。

となる。

は情報共有の便益である。 は共有範囲拡大により生じる効率改善率が逡減することを

示している。情報共有を徹底しても、その便益が小さく、情報インフラ整備コストに見合わないかもしれない。は情報共有に伴うボトルネックの発生という逆効果を表している。

5. プリウスの事例・再論

モデルの示唆を念頭に、もう一度トヨタ自動車のプリウス開発の事例を見てみよう。

本稿のモデルから、以下の情報共有の逆効果が導かれた。情報共有が行き過ぎると、共有範囲拡大による効率改善が逡減するためにその便益が情報インフラ整備コストに見合わなくなるかもしれないこと、及び 解の分布に歪みがあるときに共有範囲を広げ過ぎると特定のエージェントに問題が集中しボトルネックが発生しうることである。

プリウスの開発で使われたメーリングリスト方式は、これらの逆効果を回避し、問題の伝達を早め、効率的な解決をもたらすという情報共有の便益を上手に引き出した事例と行うことができるのではないだろうか。

メーリングリスト方式はメールを配信するサーバさえ管理していればよく、極めて安価である。範囲を限定することで、セキュリティ維持のコストも節減できる。プリウスの開発リードタイムが従来に比べ、短縮されたことを考えれば、この情報インフラ整備コストは十分にペイしたと言える。

プリウスの開発では、既存の技術をハイブリッドという新規のシステムへ擦り合わせていく必要があった。この状況は、部門間で結合する部分の解が欠落しているドーナツ型の解の分布として表現できる。このとき、情報共有による問題解決効率の改善はボトルネック発生のため、あまり大きくなかった。さらにプリウスの要素技術の新規性を考慮すると、各部門のフロンティアが広い集中型としての側面もあるかもしれない。そうすると、集中型で観察された悪影響がプリウスのケースでも起こりえたといえる。このようにボトルネックという観点から見ても、情報共有の範囲を限定されていたことには意味があった。

6. おわりに

本稿では、「プリウス」開発における情報共有を手がかりに、情報共有の範囲と組織全体の問題解決効率の関係をモデル化した。その結果、情報共有は行き過ぎない範囲においては効果的であるが、行き過ぎると逆効果が発生しうることがわかった。

徹底した情報共有による悪影響は、極めてマルチエージェント的な現象である。例えば、交通情報に基づき、各ドライバーが空いている経路を選択すれば、それらが総計されて今まで空いていた道に渋滞ができる。このように全てのエージェントに情報を与えると全体では非効率が発生する。KKMAS でも、遊園地のアトラクションの混雑情報と行列の形成がモデ

ル化され、全員が混雑情報を持つとやはり非効率が発生した〔辺見 2002〕。

それゆえ、本稿のモデルで生じた現象は、このようなマルチエージェントならではの現象であり、マルチエージェントシミュレーションという手法を採用することではじめて明らかにされるものである。

参考文献リスト

- Clark, K. B. & Fujimoto, T. (1991) *Product Development Performance*. Boston, Mass: Harvard Business School Press. 邦訳, 藤本隆宏・クラーク K.B.(1993) 『製品開発力』田村明比呂訳, ダイヤモンド社.
- 辺見和晃(2002)「来園者に優しいテーマパーク-混雑緩和問題と情報の共有」山影進・服部正太編『コンピュータのなかの人工社会』(pp.124-139). 共立出版.
- 家村浩明(1999) 『プリウスという夢 - トヨタが開けた 21 世紀の扉』双葉社.
- 碓義朗(1999) 『ハイブリッドカーの時代 - 世界初量産車トヨタ「プリウス」開発物語』光人社.
- 板崎英士(1999) 『革新トヨタ自動車 - 世界を震撼させたプリウスの衝撃』日刊工業新聞社.
- 武藤明則(2002)「新製品開発のマルチエージェントシミュレーション」
http://www2.kke.co.jp/mas/MAScommunity_output.html で入手可能。