

論文 / 著書情報  
Article / Book Information

題目(和文)	プロジェクトマネジメントにおけるコミュニケーション効率の研究
Title(English)	Study toward efficient communication in project management
著者(和文)	渡部雅男
Author(English)	MASAO WATABE
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第7752号, 授与年月日:2009年6月30日, 学位の種別:課程博士, 審査員:寺野 隆雄
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第7752号, Conferred date:2009/6/30, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

# プロジェクトマネジメントにおける コミュニケーション効率の研究

Study toward Efficient Communication in Project Management

知能システム科学専攻

Department of Computational Intelligence and Systems Science

渡部 雅男

Masao Watabe

(空白)

## 目次

概要	1
第1章 序論	2
1.1 はじめに	2
1.2 論文の構成	3
第2章 関連研究の調査とモデル化の根拠	5
2.1 はじめに	5
2.2 プロジェクト・要員・規模の定義	5
2.3 問題設定	8
2.4 研究の進め方	8
2.5 関連研究の内容	11
2.5.1 プロジェクトにおけるコミュニケーションの負荷	11
2.5.2 コミュニケーションとプロジェクトの成功	13
2.5.3 協力及び専門領域を超えたコミュニケーション	14
2.5.4 コミュニケーション効率と生産性	15
2.5.5 Face-to-faceコミュニケーションの重要性	17
2.5.6 大規模プロジェクトにおけるコミュニケーション	19
2.5.7 PMからのコミュニケーション指示	20
2.5.8 プロジェクトにおける情報交換の中身	21
2.5.9 プロジェクトマネジメントとABS	21
2.5.10 プロジェクトマネジメントと教訓	22
2.6 関連研究から導くモデル化の根拠	23
2.7 まとめ	25
第3章 シミュレーションモデルの設計と実装	27
3.1 はじめに	27
3.2 現実プロジェクトのコミュニケーションとモデル	27
3.3 シミュレーション空間におけるエージェントの配置	30
3.4 現実プロジェクト内での情報交換からシミュレーションモデルへ	31
3.5 PLエージェントの知識レベル	32
3.6 PMからPLへの指示メール出しの概念	33
3.7 PL間の知識交換	35
3.8 PMとPLの持つ数字の意味	38
3.9 研究で扱う評価尺度	38
3.10 まとめ	41

第4章 シミュレーションの実施と考察.....	43
4.1 はじめに .....	43
4.2 PLの数と指示メールの送出周期の影響.....	43
4.2.1 シミュレーションのフロー .....	43
4.2.2 シミュレーション結果と考察：PLの数の影響.....	46
4.2.3 シミュレーション結果と考察：指示メールの送出サイクルの影響.....	47
4.3 効率の良いコミュニケーション方法.....	48
4.3.1 シミュレーションのフロー .....	48
4.3.2 知識交換テーマを遡る効果 .....	51
4.3.3 シミュレーション結果と考察：知識交換テーマを遡る効果.....	52
4.3.4 PMの指示サイクルとテーマの遡及.....	53
4.3.5 シミュレーション結果と考察：PMの指示サイクルとテーマの遡及.....	54
4.3.6 ミーティング時間とテーマの遡及.....	55
4.3.7 シミュレーション結果と考察：ミーティング時間とテーマの遡及.....	56
4.4 まとめ .....	57
第5章 アンケート調査.....	58
5.1 はじめに .....	58
5.2 調査の方法 .....	58
5.3 プロジェクトにおけるコミュニケーションの負荷の大きさ.....	58
5.4 プロジェクト規模とミーティングの持ち方.....	60
5.5 まとめ .....	61
第6章 結論 .....	62
6.1 結論 .....	62
6.2 今後の課題 .....	63
謝辞 .....	64
付録 .....	71
付録1：コミュニケーション負荷に関わるアンケート .....	71
付録2：効率的なコミュニケーションに関わるアンケート .....	72
付録3：ABSのソースコード.....	74

# 概要

プロジェクトをマネジメントする際の業務時間におけるコミュニケーションの時間比率は高く、プロジェクトチーム内のコミュニケーションの効率はプロジェクトの成功に影響を与える。

プロジェクトマネジメントの知識体系である **PMBOK** に、プロジェクトにおけるコミュニケーションプロセスのベストプラクティスは書かれているものの、プロジェクト成功のためのコミュニケーションの経験則は書かれていない。また、既往研究においても、コミュニケーションを測定する有効な評価尺度を定義しておらず、有効なコミュニケーションの方法について数値的な研究がなされていない。この研究では、新たに「コミュニケーション効率」という評価尺度を定義し、エージェント・ベース・シミュレーションを行って、帰納法による結論を得るためのデータを得た。

シミュレーションモデルを、既往研究を根拠に現実を単純化して作成し、シミュレーションに供した。プロジェクトの規模、PM から PL への指示間隔、議題の数、ミーティング時間などのコミュニケーション効率への影響を調査し、それらの影響と効率的なコミュニケーションについて考察した。

本研究で得られた知見は次の通りである。1) 大規模プロジェクトは、小規模プロジェクトと比べると、コミュニケーション効率が低い。2) コミュニケーション効率を高くしたいならば、PM はプロジェクト納期の許す限り、ゆっくりと PL に指示を出した方が良い。3) PM が指示をてきぱき出す場合は、一回のミーティングで多くのテーマを議題にすると、コミュニケーション効率が効果的に上がる。4) ミーティングの回数を減らす代わりに、ミーティング時間を長くし、多くのテーマを議題にすると、コミュニケーション効率を高めることについて、相乗効果が認められる、等である。また、シミュレーションから得られた結果を、プロジェクトマネジメント関係者へのアンケート調査結果と比較し、矛盾がないことを確認した。

今後の課題として、エージェント・ベース・シミュレーションのモデルに新たな条件を設定して、知見を広げることが望まれる。また、この知見が事例研究を行うことによって明確な裏付けがなされ、公知化されることが望まれる。

# 第1章 序論

## 1.1 はじめに

近年、プロジェクト型の形態で仕事をする機会が増えてきている。プロジェクト型になってきた理由はさまざま語られているが、様々な専門性を持つ人々が集まってきて、その仕事のみの特化して分担し業務を遂行するという形態であることには変わりはない。そのため、プロジェクトマネージャやプロジェクトリーダーと呼ばれるコアメンバの統制業務の負担が大きく、プロジェクトの成功率も高くはない（日経コンピュータ[74]）。

ソフトウェア開発業界においてはプロジェクト型の業務遂行は歴史的に長い。それは業務の特殊性によるところが多いが、設計の段階で必要な人員が制作の段階で別の膨大な数の人員に作業を渡し、さらに試験行程では制作に関わらなかった人員がチェックしていた。しかし、作業が分断化されていて、ソフト制作のポイントが伝わりにくかったなどの反省があり、コアメンバが一貫して作業に携わるなどの改善が図られている。

コアメンバを配したプロジェクトであっても未だ失敗事例に事欠かない。要因はさまざま指摘されており、その中でも発注者との間のミスコミュニケーションが強く指摘されている。それらを改善するためにプロトタイピングやペルソナ開発といった形で具体物を出してギャップを顕在化させる手法や、必要要件のチェックリストを作って確認を一つ一つするなどのコミュニケーション補助手段などが作られている。

しかしながら、そのような改善にもかかわらず、プロジェクトの失敗が起きている。そこで、本研究ではプロジェクトチームの内外間のコミュニケーションギャップではなく、プロジェクトチーム内のコミュニケーション効率に問題があるのではないかと仮定し、チーム内コミュニケーションの状態により、コミュニケーション効率にどのような変化が生じるのかをシミュレーションにより確認することにした。

プロジェクトチーム内コミュニケーションを取り上げてはいるものの、フィールド調査は行っていない。実現できていればさらに有益な研究になったとは考えるが、実際のプロジェクトは秘匿性が強く、なかなか開示させてもら

えるものではないという理由の他、通常、プロジェクトに関わる人数が非常に多い上、全てのコミュニケーション現場を押さえることは事実上無理に近い。このような状況においては、初期の人工知能の研究のように研究者の内観や観察からモデルを作り、そのモデルに基づいてシミュレーションを行い、その結果を現実場面と照合していくという手法は有益であると考える。

本研究においては、上記のシミュレーションをする上で、エージェント・ベース・シミュレーションを用いた。この手法において、プロジェクトに参加しているコアメンバを全てエージェントとして置き、そのエージェントが接触あるいは非接触状況で何らかの情報を交換していることをコミュニケーションとしてモデル化している。このような単純な、モデル化であってもコミュニケーションのあり方が、コミュニケーション効率に影響を与えることを示せる。

今までの議論をまとめ、本研究ではプロジェクト型業務において、プロジェクトチームに集められてきた要員が業務を遂行していく際に、プロジェクトチーム内コミュニケーションに限って、何が影響因子であり、また、どうすれば効率の良いコミュニケーションが取れるかを探ることにした。そのために、コアメンバを中心としたプロジェクトチーム構成において、情報伝達のモデル化を行い、それをエージェント・ベース・シミュレーションした。コアメンバの人数の違いや、同時期に交換するテーマ数の違い、情報交換のサイクル、情報交換に掛ける時間などについてその特性を調べた。これらの結果から、プロジェクトチーム内のコミュニケーション効率の特性とそれを向上させるための方法について考察した。また、これとは別に、プロジェクト型業務に携わったことがある人にアンケート調査を行い、日頃の意識を調べ、モデルの考察で得た結果と比較・考察している。

なお、本研究では、著者も経験が多く、かつプロジェクト型業務を日常としているソフトウェア開発業界の実態を踏まえ考察しているが、似たようなプロジェクト組織形態で、同様なコミュニケーション形態を取る他業界のプロジェクトにおいても、この研究の結果は適用可能である。

## 1.2 論文の構成

本論文は、全6章で構成されている。



本章「序論」では、プロジェクトマネジメントに関する現状とそれに関わる問題を述べた。特に、プロジェクトの成功・失敗が、コミュニケーション効率によって左右されることを示し、この問題の分析には、エージェント・ベース・アプローチが有効であることを示した。また、本研究全体の構成を記述した。

第2章「関連研究の調査とモデル化の根拠」では、本研究の詳細な議論に入る前に、本論文で用いられる言葉の定義、研究範囲の確認や問題設定を行ないつつ、関連する研究領域を俯瞰し、研究の位置づけを明確にした。はじめに、プロジェクト、プロジェクト組織、プロジェクトの要員、プロジェクトの規模、ならびに本研究の対象範囲を定義した。次に本研究の目的としての問題設定を行い、その問題を解くための研究の進め方について記述した。その後で、プロジェクトマネジメントに関する既往研究のサーベイ結果を示し、プロジェクトにおけるコミュニケーションの特性を整理した。さらに、これらの既往研究から、エージェント・ベース・シミュレーションのモデルを単純化する根拠を見出している。

第3章「シミュレーションモデルの設計と実装」では、本研究で採用したセル型のエージェント・ベース・シミュレーションモデルについて説明している。既存研究ではコミュニケーションについての測定評価尺度が定められていなかったが、本論文で新たに導入した「コミュニケーション効率」という評価尺度について定義し、考察している。

第4章「シミュレーションの実施と考察」では、PLの数と指示メールの送付サイクルがコミュニケーション効率におよぼす影響と、効率の良いコミュニケーションを実現する方法について、シミュレーションのフローと設定条件について述べ、その結果について考察している。

第5章「アンケート調査」では、本研究の結果を、現実と対比する目的で行ったアンケート結果を報告している。報告内容は、実際のプロジェクトにおけるコミュニケーションの負荷とプロジェクトチーム内のコミュニケーションを効率化する方法についてである。

第6章「結論」では、本研究の結果をとりまとめた上で、コミュニケーション効率について、今後の更なる条件設定による研究拡張と、事例研究による本研究の裏付けが課題であるとした。

## 第 2 章 関連研究の調査とモデル化の根拠

### 2.1 はじめに

本研究の詳細な議論に入る前に、本論文で用いられる言葉の定義、研究範囲の確認や問題設定を行ないつつ、関連する研究領域を俯瞰し、研究の位置づけと研究の進め方を明確にする。

問題設定を踏まえて関連研究を調査し、プロジェクトのコミュニケーションに関する関連研究及びプロジェクトの ABS に関する過去の研究結果について分類する。分類項目は、「プロジェクトにおけるコミュニケーションの負荷」、「コミュニケーションとプロジェクトの成功」、「協力及び専門領域を超えたコミュニケーション」、「コミュニケーション効率と生産性」、「Face-to-face コミュニケーションの重要性」、「大規模プロジェクトにおけるコミュニケーション」、「PM からのコミュニケーション指示」、「プロジェクトにおける情報交換の中身」、「プロジェクトマネジメントと ABS」、「プロジェクトマネジメントと教訓」である。

また、分類された関連研究からモデル化の際の単純化の根拠について述べる。

### 2.2 プロジェクト・要員・規模の定義

PMBOK ガイド [7] は、プロジェクトを次のように定義している：「プロジェクトとは、独自のプロダクト、サービス、所産を創造するために実施される有期性の業務である。」 また、PMI 東京支部ホームページ[23]では、「プロジェクトとは、明確な始まりと終わりがある成果物またはサービスを生み出すための有期的な業務である。」と記述されている。

そのようなプロジェクトの中で、本研究は、連携性の高いソフトウェア開発プロジェクトを対象とする。連携性が高いとは、図 1 に示すように、プロジェクトの各フェーズを担当するメンバー間、及び同じフェーズを担当するメンバー同士が縦と横の連携を図って遂行することを示す。プロジェクト期間を通して、単に担当組織を流れてゆくようなプロジェクトは対象としていない。連携性の高いプロジェクトほどプロジェクトチーム内部のコミュニケーション頻度が高く、コミュニケーション効率の良否がプロジェクト遂行効率を

左右する。

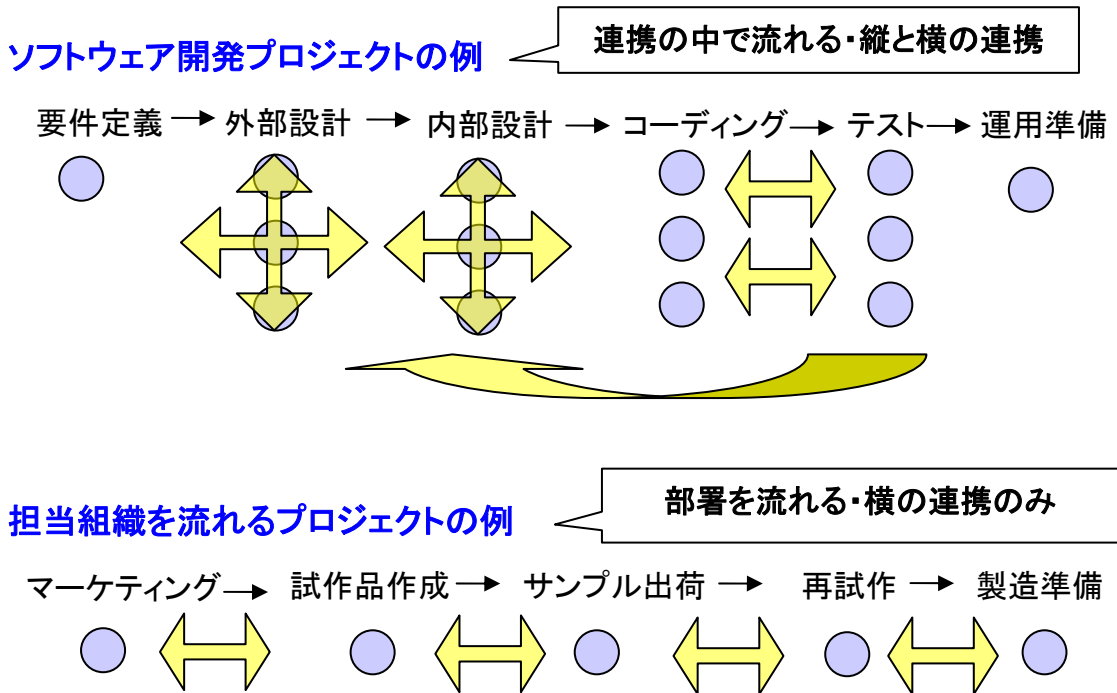


図1 連携性の高いプロジェクトとは

プロジェクトの迅速かつ効率的な運営には、プロジェクト遂行中に並走する様々な専門要素分野の作業のインターフェースが、単純になるように進めることが望ましい。ソフトウェア開発に於いてもプログラムのモジュールの独立性を高め、ソフトウェアの複雑性を回避し、開発及び保守の容易性を増すための努力が行われている。しかし、専門要素分野の作業のインターフェースが高められるのは、ソフトウェアであれば、コーディングの部分であり、建設系のプロジェクトであれば、施工の段階のことである。それを実現するためには、どうしても上流の設計工程に於いて、専門要素分野を担当する専門家間の十分な連携により、下流工程でのインターフェースが少なくなるように段取りすることが必要になる。この研究はこの上流部分を対象としており、ソフトウェア開発であれば、コーディング期間でのコミュニケーションの複雑性を想定するものではない。

また、本論文では、プロジェクト遂行に携わるプロジェクトチーム要員を下記のように定義している。

- 統括プロジェクトマネージャ（統括 PM）：必要に応じて、経営的観点からプロジェクトマネージャを指導、支援する。

- プロジェクトマネージャ (PM) : プロジェクトの運営を統括し, プロジェクトチームをまとめる.
- プロジェクトリーダー (PL) : プロジェクト内の専門領域のプロジェクトメンバをまとめる専門家のリーダー. 他の専門領域との情報交換と調整を主に行う.
- プロジェクトメンバ (M) : プロジェクト内で特定の専門領域の業務を自己完結的に近い形で行う者.

実プロジェクトメンバには, この区分け以外のスタッフ的プロジェクト要員が存在するが, プロジェクト遂行において支配的な役割を果たすわけではないのと, プロジェクト毎に状況が異なるので, モデル化から除外する.

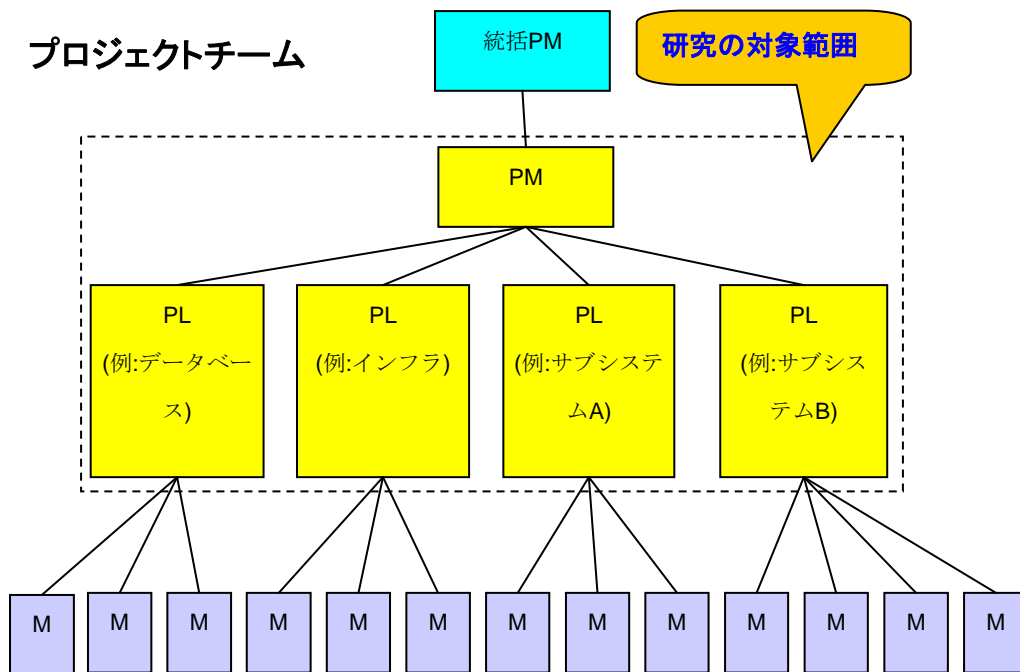


図2 ソフトウェア開発プロジェクトチームの標準的構成

一般に, プロジェクトの規模は, プロジェクトに関わる要員数, 工数, 予算金額などで定義される. 例外的なプロジェクトでなければ, プロジェクトの規模とそこに关わる要員数はほぼ比例すると考えてよい. 仲谷[6]の論文ではソフトウェア開発における大規模なプロジェクトは, 百人から数百人のメンバから構成されるとしている. 筆者の大手システムインテグレータのシステムインテグレーション部門に対するヒアリングによると, 2001年, 9ヶ月間のプロジェクト期間, 数十億円強の受注金額, 情報通信系のソフトウェア開

発事例（これは大規模プロジェクトである）における技術要素ごとの専門チームのピーク時の数は、15 チームであった。チームごとに PL がアサインされたので、PL が 15 人いた。

本論文では、「プロジェクトの規模」を下記のように定義する。

- 大規模プロジェクト：PL の数が 11 人以上のプロジェクト。
- 中規模プロジェクト：PL の数が 5 人～10 人のプロジェクト。
- 小規模プロジェクト：PL の数が 4 人以下のプロジェクト。

ソフトウェア開発のプロジェクトチームは、一般に図 2 のような組織で遂行される。超大規模プロジェクトの中には、PL 部分が二層構造になっているものがまれにある。コミュニケーション効率を考察することにおいて、例外的に行われる二重構造をここでモデル化の範囲にすることはしない。また、プロジェクトチームの組織は、プロジェクトの開始、進行、終結をたどる時間軸で、ゆるやかに変わってゆく、この場合、図 2 に示す組織の構造形態は変わらずに、PL の人数や PL の率いるグループの役割が変わってくる。この時間軸による推移は本研究のモデル化では扱わず、ピーク時の組織をモデル化することによって、コミュニケーション効率の傾向を見る。コミュニケーション効率の絶対値を知るのではなく、傾向を見ることに限定すれば問題ない。

## 2.3 問題設定

本研究では、次のような因子が、コミュニケーション効率にどのような影響を及ぼすのかを調査し、コミュニケーション効率を高めるにはどうしたら良いのかについて考察する。

- プロジェクトの規模（PL の数）
- PM から PL への指示間隔
- ミーティングにおける議題の数
- ミーティング時間

## 2.4 研究の進め方

コミュニケーション・マネジメント研究会[27]によると、近年、IT プロジェクトは大型化し、プロジェクトにおけるコミュニケーションは、より複雑化してきていると言われる。Myers [14]によると、複雑なソフトウェア開発で

は、プロジェクトメンバー間の緊密な協力が必要であり、各機能部門から選抜されたメンバーがコミュニケーションしながら相互依存関係を持っている。また、高木 [46] は、プロジェクトは複雑でダイナミックな人間中心の解放系システムであり、このような複雑な連携（コミュニケーション）を、演繹的な数学的手法でモデル化することは難しいと言っている。このようなプロジェクトメンバーが複雑に相互作用を及ぼす状況を研究する場合には、Axtell [72]が言うように、Agent-Based-Simulation (ABS) が有効である。

プロジェクトにおけるコミュニケーションは、相手先・カテゴリー（情報の伝達、調整、交渉）、情報伝達の方法・頻度・内容は、状況と時間軸によりダイナミックに変化するために、待ち行列などによる数式モデルにより表現することは難しい。また、実際のプロジェクトでは、多くのプロジェクトメンバーが、同時に、多くの場所で、個別もしくは集合で相互に情報交換をしており、ここからコミュニケーション効率の統計データを取り出すことも時間と費用が非常に掛かる。これが帰納法によるこの分野の研究が遅れている理由だと思われる。プロジェクトチーム内のコミュニケーションはダイナミックなゆえに、単純に数式モデルに置き換えることは出来ない。一人ひとりのエージェントに単純化した形で行動ルールを組み込み、その相互作用により示唆を得る ABS は、「シミュレーション効率」を研究するという本論文の目的をかなえる道具としては、ふさわしいものである。

ABS は、米国のサンタフェ研究所による有名な「複雑系」研究の過程で発展してきたもので、コンピュータ上に設定した複数のエージェントに同時にルールを実行させ、エージェント間の相互作用から、社会現象などを観察する方法である。典型的な ABS のモデルでは、コンピュータ画面上にセルに区切られた空間を設定し、その画面上に配置した複数のエージェントをランダムに動かし、相互作用をさせて、それを観察する。エージェントにあらかじめプログラムを組み込むだけでなく、空間にもさまざまな条件を組み込むことが可能である。

Axelrod[12]は、シミュレーションは科学を行う第三の方法であると定義した。演繹法と同じく、はじめに明確な仮定を置くが、演繹法と異なり、理論的な証明をするのではなく、帰納的な分析をするためのデータを作り出すものである。典型的な帰納法が直接現実のデータを測定することと異なり、ABS では、データは厳密に決めたシミュレーションルールから生まれる。演繹法は仮定から得られる結果を見つけ、帰納法はデータから特定のパターンを見

出すために用いられると言われる。このように、ABS のモデルは、得られた特定のパターンから洞察をするために用いられる。

また、Axelrod[12]は、ABS は、特定の実証可能な正確なモデルを提示することが目的ではないと言っている。それよりも、現象の基本的なプロセスの理解を深めることを目的としている。そのために KISS(Keep It Simple, Stupid) 原理に従う必要があり、モデル化する対象が複雑であればあるほど、モデルは単純である必要があると述べている。モデルを複雑に作ると、シミュレーションの結果は、モデルの仮定ではなく、その複雑性に左右されるからであるとも言っている。最近の ABS による分析では、現実を精緻かつビジュアルなモデルに作りこみ、現実との比較や予測に使われているのも事実であり、全てに KISS 原理を当てはめるわけには行かない。しかし、現実をモデル化すると非常に複雑となるものについては、モデル構造あるいは条件パラメータの設定によって結論が大きく異なることがある。このことが、設定した仮定に意図的に有利な結論を導くようなモデル設定あるいはパラメータ設定しているのではとの疑念を抱かせる。ここに Axelrod の KISS についての論点があり、ABS については、シミュレーション対象を詳細にモデル化するのではなく、要点をおさえた単純な概要モデルにより、傾向を洞察するところに意義があるとするものである。ABS は、現実世界のプロセスを単純なモデルで置き換え、得られた結果を現実世界の証明として用いることを目的とせず、あくまで現実世界の複雑なプロセスの大きな流れを洞察する目的に限定する場合には、数値的な評価が出来るメリットがある。

今回のシミュレーションモデル作成にあたって参考にしたモデルは、図 3 に示す Axelrod [24]の「文化の伝播（流布）モデル」である。これは、ひとつの空間（世界と考えても良い）を  $10 \times 10$  のセルに分け、全てのセルに合計 100 のエージェント（村と考えても良い）を配置させる。各々の村は独自の文化を持つが、それは、0~9 の値を取る 5 種類の文化要素を持つ。例えば、78295 のように 5 桁の数字で表わされる文化要素列を持つことになる。これらのエージェントは、隣り合っているエージェントの中から、自分の文化に近い文化要素列を持つエージェントとコミュニケーションを行い、文化が伝播する。文化の伝播によって、同じ文化の領域が局所的に形成される。Axelrod はこのモデルにより、局所的な収斂がグローバルな分極化を引き起こしうること、さらに、文化の様々な特徴同士の相互作用が社会的影響のプロセスを形成しうることを示している。これにより、文化には広まる特性があるが、一つには収束しない（文化が安定した領域が一定の広さ以上に大き

くなると、安定領域の数が減ることである。)ことを解明した。本研究では、このモデルの文化要素の伝搬を、プロジェクトの PL の持つ専門知識の伝搬に置き換えて考えた。本研究は、プロジェクトにおける接触コミュニケーションを Axelrod のシミュレーションと同じレベルの単純さでモデル化し、同じく数値列で知識の伝搬効率をシミュレーションするものである。Axelrod のシミュレーションと異なる点は、エージェントの数の増加とともにその知識の数を増やしていることであり、Axelrod のシミュレーションにおいて、文化の種類を村の数にリンクして増やすことに相当する。

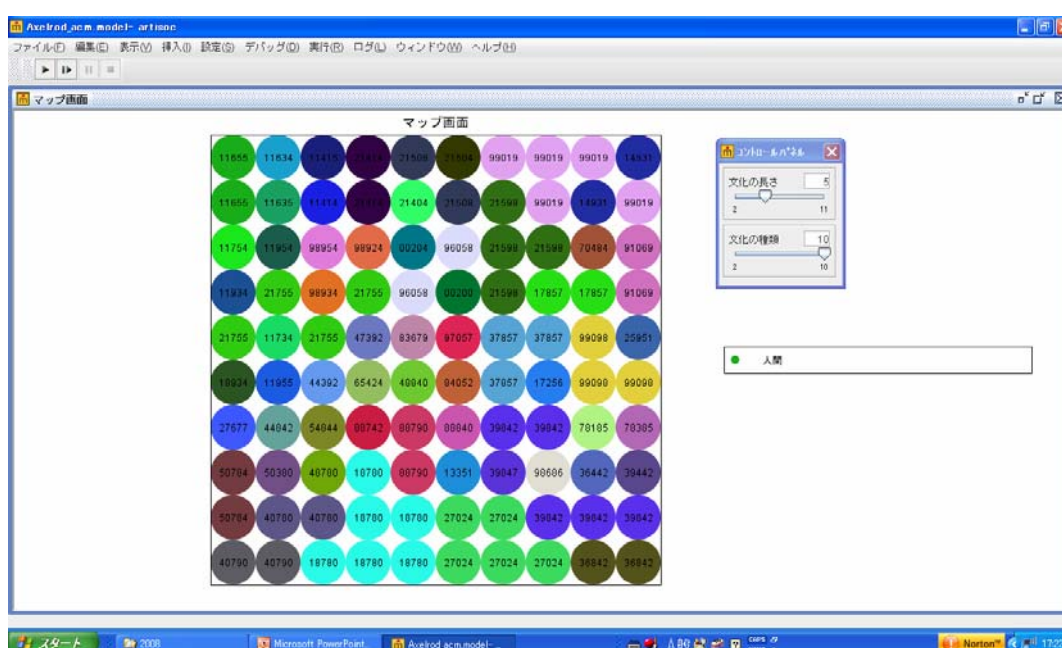


図 3 構造計画研究所 MAS コミュニティー サンプルモデル 「Axelrod の Adaptive Culture Model」 から

## 2.5 関連研究の内容

筆者は、問題設定を踏まえ、関連研究を行った。以下に、プロジェクトのコミュニケーション効率に関する関連研究、及びプロジェクトの ABS に関する過去の研究結果について分類した。

### 2.5.1 プロジェクトにおけるコミュニケーションの負荷

ブルックス[8]は、コミュニケーションを取りながらこなす仕事においては、本来の仕事にコミュニケーションの労力が加わることを述べている。Gillard



ら[18]は、プロジェクトマネージャは、多くの時間をコミュニケーションに使うと述べている。関根 [38]は、プロジェクトマネージャの仕事時間の 90% 近くはコミュニケーションに費やされていると言及している。秋山 [40]は、プロジェクトの成功において、効果的なコミュニケーションは不可欠であり、プロジェクトマネージャは、その時間の 90%をコミュニケーションに費やすといわれると言っている。遠藤 [34]も、プロジェクトマネージャは、プロジェクトのワークロードの 90%近くを「人と人とのかかわり」に費やしていると言っている。また、竹久 [66]は、「プロジェクトマネジメントは、PMBOKで言う九つの知識エリアのプロセスが相互に PDCA を回しながら進んでいく。この中でも重要なことはプロジェクトのステークホルダの巻き込みであり、コミュニケーション・マネジメントの問題であると」述べており、優秀なプロジェクトマネージャは自分の時間の大部分（80%以上）をコミュニケーションに割いているとも言っている。

Carvalho[71]は、従業員約 2,500 人の、複雑で大規模なプロジェクトを多数扱っている IT 企業で調査をした結果を示している。それによると、コミュニケーション時間の内、ミーティングが 20.2%、インフォーマルコミュニケーションが 9.9%を占めている。両方を加えるとおおよそ 30%となる。Root[69]は、オフィスに関する研究で、オフィスワーカーのアクティビティの内 65%が人と人とのコミュニケーションに使われ、32%が対面コミュニケーションであると報告している。

東ら[3]は、コミュニケーションの負荷がソフトウェア開発プロジェクトで高いのは、複雑な相互関係を持つシステム的な作業であるからと分析している。更に、ブルックス[8]は、ソフトウェア開発は複雑な相互関係における作業の遂行であり、増員しても、コミュニケーションの労力は、各プロジェクトメンバに割り振られた本来の作業時間の短縮分をすぐに上回ると述べている。

プロジェクトにおけるコミュニケーション負荷は大きく、プロジェクト規模が大きくなるほどさらに大きくなる。山崎ら[60]は、大規模開発ほど、プロジェクト関係者が増加し、プロジェクトメンバにとって、コミュニケーションを行う相手の数は指数関数的に増大すると書いている。仲谷[6]も、プロジェクトの人数が多くなると、作業のほとんどがコミュニケーションに費やされるといっても過言ではないと指摘している。理由として、ブルックス[8]が、コミュニケーションに必要な労力は、プロジェクト要員が  $m$  人いれば、 $m(m-1)/2$  に比例することを上げている。しかし、この式もプロジェクトにお

けるコミュニケーションの複雑性を十分に表現するに至っていない。

これらの研究は、プロジェクトにおけるコミュニケーション負荷はかなり高く、それは、プロジェクト規模が大きくなるほど顕著になることを示唆している。コミュニケーションは、プロジェクトマネジメントの成功や効率に影響を与える因子である。これらの論文には、実際の測定に基づかない経験的数値が含まれている。しかし、その数値自身をこの研究で使うわけではなく、コミュニケーション負荷が高いことを、プロジェクトにおけるコミュニケーション効率研究の意義を裏付けるものとして引用するのは問題ない。

## 2.5.2 コミュニケーションとプロジェクトの成功

小松 [35]、治田 [37]、松島 [41]、久井 [56]らは、プロジェクト不成功の原因に、よくコミュニケーションの不良があげられると報告している。コミュニケーション・マネジメント研究会[59]は、情報システムプロジェクトの成功率は依然として高くなく、その失敗原因の一つは「不適切なコミュニケーション」にあるとしている。松島 [64]は、プロジェクトが失敗したとき、あるいは、途中で大きな障害が発生したとき、そこにはコミュニケーション上の問題があったと常に言われ、コミュニケーションの重要性が繰り返し指摘されてきたと述べている。Pinto ら[19]は、研究結果として、コミュニケーションがプロジェクトの成功に大きく影響することを見出している。伊東[2]も、コミュニケーションの有効性は、プロジェクト組織の目標達成と正の相関があると論じている。

具体的な事例をあげているのは次のような文献である。藤田 [28]は X 社の IT プロジェクトの事例をあげ、コミュニケーションの不足もプロジェクトが失敗する大きな原因となることを示している。また、藤田[28]は請求システム更新プロジェクトの事例で、適切にコミュニケーションを取ることによって、一旦破綻しかけたプロジェクトを復旧することができたと述べている。実森 [39]は、ソフトウェア開発の事例から、コミュニケーション技術は、プロジェクトを成功に導くための鍵になると言っている。治田 [43]も、プロジェクト実施においてはコミュニケーションをいかにして適切に実施するかが、成否を分ける重要な要因であることを、事例をもとに述べている。

分析的には次のような文献がある。コミュニケーション・マネジメント研究会[27]によると、IT プロジェクトの特徴は、IT プロジェクトの可視性の悪さ

とプロジェクト管理プロセスの未熟さに起因しているところが大きい。そのため、すでに不確実性やリスクを抱える IT プロジェクトの遂行にあたっては、それを補完する意味でコミュニケーションへの依存度は極めて大きいと言っている。コミュニケーション・マネジメント研究会[65]は、コミュニケーションは社会的な活動において欠かすことができないが、行動に伴って自然に発する行為として殆ど無意識的に行われることが多い。したがって、プロジェクトという一定の前提条件、制約条件下における活動であっても、マネジメントという概念が希薄であると分析している。

これらの研究は、複雑なプロジェクトほど、コミュニケーションのあり方がプロジェクトの成否に大きく影響することを示している。しかしながら、これらの既存研究は事例をヒアリングして知見を導いたものが多く、なぜこのようなことが言えるかについて根拠を明確にして分析しているものはほとんどない。

### 2.5.3 協力及び専門領域を超えたコミュニケーション

森川 [32]は、プロジェクトマネジメントでは、各現場のプロジェクト管理者が、それと関連する別のプロジェクト管理者との間でそれぞれ打ち合わせを行いながら、全体のプロジェクトを進めていくというスタイルになり、この方法を日本では「すり合わせ」という言葉を使って表すと言っている。この方式の長所は、各現場が他の現場で行っている事を相互に把握し、それを通じてプロジェクト全体を広い視野で捉えることが可能であるという点であり、これによって各部門の細かな変更にも柔軟に対応できるようになることだと述べている。

Pinto ら[19]は、部署を超えた情報共有の欠如は、プロジェクトの失敗につながることを述べ、部署を超えた協力のレベルの高いプロジェクトが、より成功するという研究結果を示している。Woodward [15] は、理想的なコミュニケーションパターンを図で示し、PM とその下にある全てのグループが緊密にコミュニケーションすることを理想としていると述べている。

事例を紹介し、Zimoch [13] は、プロジェクトを進め、問題を解決するには、協力、コミュニケーション、チームワークが大切であるとしている。大黒 [63] は、実失敗プロジェクトの事例報告をしている。財務、文書、旅費、基盤の4つのサブシステムによって構成された大規模プロジェクトで、スケジュー

ル遅延, データ不整合, 回復作業に大量の工数投入などのトラブルが発生したケースである. 問題の原因の一つとして, サブシステム間のインターフェースのコミュニケーション不足が挙げられている. 部門間の調整力は弱かったとのことであり, 他システムについての理解不足から生じる誤解やミスを軽減するために, 各業務の勉強会を実施して各々の業務の特徴を把握するようにしておけばよかったはずだと言っている. このように, 部門間の調整不足はプロジェクトのトラブルの原因になるので, ミーティングによる情報交換の必要性を強調している.

分析的な論文としては次のようなものがある. Nakatani ら[20]は, プロジェクトにおいて協同して作業するためにはコミュニケーションが欠かせず, その理由として, 複雑なプロジェクトでは, 正しい判断をするために, プロジェクト内のさまざまな知識を総合する必要があると述べている. Myers[14]も, ソフトウェア開発は元来複雑で不確実性が高いので, 成功するためにはプロジェクトのメンバが緊密に協力しなければならないと書いている. 伊東[2]は, 多くの専門家が集まるプロジェクトでは, メンバが異なる経歴と専門を持つことから, 異なる文化を持っており, コミュニケーション管理が難しいことを上げている. Thomas ら[70]は, プロジェクトの成功のためには, 効果的なプロジェクトチーム内のコミュニケーションが大切な理由として, チームメンバのバックグラウンドの多様性を挙げている. Turner[17]は, プロジェクトが大規模になるほど, 多くの専門家によってプロジェクトチームが構成され, 互いの領域を超えたコミュニケーションを促す必要を述べている. 込山ら[45]は, プロジェクトでは, 永続型組織と比較し, より短時間で, より多くの情報交換が必要である. すなわちコミュニケーションが重要な役割を果たしていると分析している.

これらの論文では, コミュニケーションが, 特に各チームメンバの専門領域を超えて行われることの重要性を示している. これを本論文のシミュレーションモデル設定のよりどころとした. 具体的には, 専門領域を代表する PL の知識交換を ABS でモデル化している.

#### 2.5.4 コミュニケーション効率と生産性

横島ら[42]は, プロジェクトチームのパフォーマンスを高めることは, チームビルディングとコミュニケーションに負うところが大きいと報告している. 仲谷[6]は, コミュニケーションの不良がソフトウェア生産性を阻害する

重要な要因であることを確認している。東ら[3]は、ソフトウェア開発の作業は、複雑な相互関係をもつ体系的な作業であり、プロジェクトメンバー間のコミュニケーションの労力は多大なものとなると論じている。そして、プロジェクトメンバー間のコミュニケーションを複雑にしないために、チームをいかに編成するかが、ソフトウェア開発マネジメントにとって重要であると述べている。会田 [50]は、マンパワー的には同能力でも、「組織」と「コミュニケーション」の運営の仕方によって生産性にかなりの差がでる場合が多く、「プロジェクト能力 = 組織能力 x コミュニケーション能力」としている。

ブルックス[8]は、作業者を増やしても、効果が上がるのは、コミュニケーションを図らなくても仕事が分担できる場合だけであると述べている。また、後藤ら[4]は、ソフトウェア開発に関わる人数とコミュニケーションの量について、期間の短縮をはかるために、開発の途中で人数を増やしても期間を縮めることにはならず、極端な場合、逆に遅れが発生することさえあり、その理由として人数が増えるとコミュニケーション量が大幅に増えるからであると説明している。

江熊 [36]は、プロジェクトチームの生産性に影響を及ぼす要因として、プロジェクトの規模、メンバー間のコミュニケーションなどを挙げ、プロジェクト規模が大きくなればなるほど、メンバー数と生産性の関係は理想的な直線から低下する方向に外れてゆくと述べている。理由として、複数人数で作業を実施する場合には、一人で作業を行う場合に比べ、作業者間での作業範囲や作業レベルの打ち合わせなど、意思疎通のための時間が必要となるとしている。このための時間は生産性を減少させる。作業者間でのコミュニケーションのための時間は、プロジェクト規模に比例して増加すると述べている。服部ら[47]は、平均要員数の大きいプロジェクトは、小さいプロジェクトよりも生産性が低い傾向が報告されているが、この傾向を科学的に確認する分析方法は、まだ十分に検討されているとは言えない。要員数が小さいプロジェクトは、大きいプロジェクトよりも生産性がばらついているように見えるという傾向もあると報告している。また、小さいプロジェクトでは、能力の高い技術者の有無が生産性に大きく影響するのに対し、大きいプロジェクトでは、個人の能力による生産性の向上は難しいなどが想定されると述べている。

ISBSG [21]は、ソフトウェア生産のプロジェクトにおいて、チームサイズ(チームの要員数)が大きくなると、どのようなプラットフォームやどのプログ

ラミング言語を用いても、生産効率が著しく落ちると指摘している。チームサイズのインパクト、特に大規模チームにおける好ましくないインパクトはプロジェクトの計画時に考慮しなければならないとも述べている。

Thomasら[70]は、今まで、プロジェクトチーム内のコミュニケーションの重要性について多くの研究が指摘してきたが、研究の成果はあまり上がっていない理由の一つを、コミュニケーションの有効性に関して測定する評価尺度が存在しないことであると論じている。

これらの研究は、プロジェクトチーム内のコミュニケーションが、そのプロジェクトの成功に影響するだけでなく、生産性つまりコストに大きな影響を与えていることを示している。ISBSG [21]は、その生産性を過去の実績データから数値として示しており、プロジェクト規模つまりコミュニケーションの複雑さ、と生産性の関係を明らかにしている。

### 2.5.5 Face-to-faceコミュニケーションの重要性

町田 [33]は、コミュニケーション不足を引き起こす原因の一つに、電子メールやFAXなどに頼りすぎることを挙げている。小松 [35]は、管理層では、文書のやり取りだけでは解決が困難な問題を解決するために、対面コミュニケーションによる会議が必要になると述べている。関根 [38]は、プロジェクトでは、ミーティング、メールやグループウェアを利用した文書管理だけでは適切なコミュニケーションが十分取れないと言っている。嶋野 [58]は、通常なら2~3年を要する複雑なソフトウェア開発プロジェクトを約1年で完了したPMは、「プロジェクトの推進には様々な部門や関係者との適切なコミュニケーションが欠かせない。このプロジェクトでは、公式のコミュニケーション以外に、非公式のコミュニケーションルートを確立した。」と言っている。

仲谷[6]は、プロジェクトにおいては、文書によるコミュニケーションでは十分な情報が伝わらないことを指摘している。また、調査の結果として、連絡会議や進捗報告では本当に知りたい情報は得られず、頻繁なコミュニケーションを持ち、じっくり話し合うことが有用だと報告している。Michalski[16]は、プロジェクトにおける効果的コミュニケーションには、face-to-faceが効果的であると述べている。実森 [39]は、SIベンダーのPL担当者の声として、情報共有のためにメールを使う方法もあるが、「直接対話が望ましい。メー

ルではちゃんと読んでもらったか分からない。表情や声色など、五感をフル活用できる分だけ過不足や誤解がなく、情報が伝わりやすい。」を記載している。秋山 [40]は、コミュニケーションの目的に応じて、言語によるもの、非言語によるもの及び書面によるものを組み合わせる必要がある。非公式なコミュニケーションでは、プロジェクトに関する有益な情報が得られる。プロジェクトメンバと話をすることで、報告書では分からない、雰囲気やモチベーションなどが分かる場合もあると報告している。山本ら[57]は、ネットワークを介して行うコミュニケーションに比べて、直接対面して行うコミュニケーションの方が柔軟性に富んでおり、有意義で集中的かつ効率的な会合を得たとの印象を受けやすいことを報告した研究はすでに散見されるとしている。松島 [64]は、決められたタイミングにおける決められた内容のコミュニケーションも必要であるが、人と人との会話の中から創発的なコミュニケーションが認められることも多い。形骸化しやすく、官僚的になりがちなルーチン的なコミュニケーションを、本来の価値にもどすのはアドホックなコミュニケーションの重要な役割であると述べている。電子的ツールを、face-to-face が補完するという認識で、コミュニケーション・マネジメントをデザインすべきだと指摘している。コミュニケーション・マネジメント研究会[27]は、ITプロジェクトのコミュニケーションの特徴に、設計情報の数値化や可視化が難しく、伝達しにくい情報を扱うことを指摘しているが、これが face-to-face の有効性の理由であろう。

Pinto ら[19]は、協力体制が密接なプロジェクトチームは、非公式なコミュニケーションをうまく使っていると述べている。仲谷[6]も、公式のコミュニケーションは、日常会話で補われるとも報告している。Pinto ら[19]は、協力関係の良いプロジェクトでは、計画されたミーティングではなく、廊下での立ち話なども有効に使っていると例示している。中谷ら[49]は、近年、プロジェクトにおける交わりのなコミュニケーション、いわゆるインフォーマルコミュニケーションの重要性が認識され始めている。しかし、組織におけるインフォーマルコミュニケーションの研究は少ない。評価方法も確立していないと述べている。Michalski[16]も、プロジェクトにおけるコミュニケーションには、face-to-face が効果的であると述べている。Pinto ら[19]は、互いの接触を促進するために、オフィスのいたるところに小さな会議室を設ける企業があることを紹介している。

仲谷[6]は、プロジェクトにおいて、作業場所が近い人のノウハウをより受け取ることを示し、互いの距離が近いことは、かゆいところに手が届くコミュ

ニケーションに有利であると述べている。岡村[1]は、超大型コンピュータソフト開発プロジェクトにて、コミュニケーション管理として、相対での会話ができる状態を確保するために、1,000名を越える開発要員を、15分以内の距離に、700名収容できる2棟に収容した事例を紹介している。高木 [46]は、鉄道の改札で使われているパスネットの開発プロジェクトの事例で、成功要因として、設計作業のロケーション一体化によりコミュニケーションがうまく行えたことを挙げている。

これらの論文は、プロジェクトのコミュニケーションにおいて、接触によるコミュニケーションが重要であることを述べている。本論文のモデル構造は、この接触コミュニケーションをモデル化している。

## 2.5.6 大規模プロジェクトにおけるコミュニケーション

長谷川 [44]は、近年、大規模・短納期システム開発が増加していると報告している。近藤 [31]は、最近、失敗プロジェクトが増加傾向にある。原因の一部に、開発規模肥大化、短期開発化があると言っている。大型難プロジェクトの需要は増加傾向にあり、これから大型難プロジェクトの成功確率を上げることが必要だと述べている。コミュニケーション・マネジメント研究会[27]は、近年、ITプロジェクトはさまざまな環境変化に直面し、プロジェクトの大型化・複雑化に伴い、多くのステークホルダが関与する複雑な組織体制となっている。従って、コミュニケーション自身も複雑化してきており、もはや従来型の成り行きや暗黙のコミュニケーションの仕組みでは対応が難しくなってきたと述べている。ブルックス[8]は、バベルの塔の建設を例に、大規模プロジェクトにおけるコミュニケーションの重要性を指摘している。会田 [50]は、プロジェクトが大規模になればなるほど「組織とコミュニケーション」が成功要因に大きく関わってくると述べている。

石倉 [61]は、大型生産設備プロジェクトを、納期を短縮して成功させた体験として、朝一ミーティングの徹底などコミュニケーションの徹底と効率化を図ったと報告している。倉谷 [68]は、納期が厳しいソフトウェア開発プロジェクトで、SEは自分の担当領域についてまとめるので手一杯で、サブシステム間の話し合いをするだけの余裕がなかった。開発チーム内でも軋轢が生まれていた。メンバがバラバラに動いていた。数々の問題点を開発チーム全員で共有することでチームの改善ができたと言った短納期プロジェクトの難しさを報告している。



長谷川ら[56]は、会議はメンバの増加とともに、長時間に及ぶことがある。情報量が多くなった原因は、開発期間の短縮化、多数の専門家の相互協力化などがあげられ、プロジェクトに参加する多数のメンバ同士が「情報共有」し協調することにより、大型のプロジェクトが短時間で完成すると述べている。岡田 [51]は、失敗事例について分析し、その教訓を生かす活動の対象として、短納期化も対象となっていると言っている。青木 [29]は、短納期プロジェクトでコミュニケーションに問題があるとカオス様現象が発生することが知られていると述べている。

仲谷[6]は、大規模プロジェクトになると、情報が全体に正確に速やかに伝わりにくい。また全体の情報が担当者まで行き渡らないため、全体を見通した開発が出来ない問題を指摘している。その理由として、大規模なシステム開発になると知識や情報は複数の人間に分散するからだとしている。一つの問題を解決するにも複数の人間から関連情報を収集しなければならなくなり、コミュニケーションの数が増え、コストがかかる。また、知識の分散が進むとだれがどのような情報を持っているかという情報もわからなくなり、必要な情報が得られなかったり、必要な人間へ情報が届かなかつたりするという問題も発生すると述べている。結果として、大規模なプロジェクトでは、小規模なプロジェクトに比べコミュニケーションにかかるコストが膨大なものになると述べている。

これらの既往論文は、プロジェクトチームにおけるコミュニケーションがその規模によって変わることを示しているが、定量的には裏付けてはいない。

### 2.5.7 PMからのコミュニケーション指示

治田 [43]は、プロジェクトのスコープについては、全メンバと共有化を図るため、一方向の情報伝達会議でコミュニケーションを取ると述べている。松澤ら[55]は、学生によるプロジェクト実験で、大幅なスケジュール遅延を生じたプロジェクトチームの遅延原因は、このPMがプロジェクト内コミュニケーションの問題をかかえていたことにあった。このプロジェクトチームのPMはメンバの自律的な活動を期待していたが、学生は要件定義の目的がつかめず、指示を待ったことが原因であると報告している。

大坂 [62]は、大型建設プロジェクトのコミュニケーションにおいて、情報の

一元管理と検索は不可欠で、その体系的な管理には **WBS** が使われる。IT プロジェクトでもプロジェクト規模、情報交換量、コミュニケーションコストによっては使えると述べている。山崎ら[60]は、大規模開発ソフトウェア開発において、プロジェクトの進行に合わせてスコープを順次明確化し、共通認識化する事例を挙げている。

これらの報告は、PM の指示出しの重要性と **WBS** の項目（スケジュールの項目と考えると良い）の進捗に基づく指示出しの存在を表している。本論文では、**WBS** の進捗（スケジュール進捗）に基づいて、PM が PL に対して一方向的に、PL 同士がコミュニケーションすることを促すモデルとしているが、これらの報告はその根拠となっている。

### 2.5.8 プロジェクトにおける情報交換の中身

仲谷 [6]は、コミュニケーションの評価を「個々のメンバ間の問題（ミクロな評価）」と「プロジェクト全体の問題（マクロな評価）」の2つのレベルから行うべきとしている。「マクロな評価」はプロジェクト全体のコミュニケーションの傾向を示すものであると定義している。

また、Turner[17]は、プロジェクトエンジニアは良きコミュニケーターであり、コーディネーターである必要性を述べている。プロジェクトメンバのコミュニケーションには、調整（コーディネーション）と交渉を含むことを示唆している。

本研究では、上記で示すミクロな評価の内、単なる情報交換及びコーディネーションと交渉の結果をひっくるめて伝達することをモデル化することにした。

### 2.5.9 プロジェクトマネジメントとABS

長瀬 [30]は、プロジェクトチームでは、各機能部門から選抜されたメンバが複雑な相互依存関係にあるため、職能横断的協調が必須である。相互依存が大きいほど協力の努力の必要性が高まると述べている。高木 [46]は、ソフトウェア開発プロジェクトの失敗要因あるいは成功要因に関してはさまざまな調査や研究が行われている。しかし、いずれの調査・研究も事例に基づいた現象面からの帰納的アプローチが多い。これは、プロジェクトが複雑でダ

イナミックな人間中心の解放系システムであり、解析的な扱いがきわめて困難だからであるとしている。

相良ら[25]は、プロジェクトの組織構造とプロジェクトを構成する作業特性の違いがプロジェクト全体の作業効率にいかなる影響を付与するかを ABS で検討している。

Axtell[72]は、今日、数学的な社会学者の間では、数学的な用語のモデルを述べて、次にそれを分析的に解決することを試みるというのは一般的なやり方である。しかし、相互作用によって動くエージェントが生み出す結果のように、証明できる数学的モデルが明らかでないか、扱いにくい場合、エージェント・コンピューティングは唯一有効な分析手段だと言っている。また、出口 [67]は、シナリオの時間軸に沿った比較を可能としてくれるモデルは多くない。ABS はその重要な可能性の一つであると述べている。

プロジェクトにおけるコミュニケーションは、相互依存的でダイナミックなゆえに、数学的モデルを立てにくい。これが、本研究で ABS を用いた根拠となっている。

## 2.5.10 プロジェクトマネジメントと教訓

遠藤 [34]は、このコミュニケーション・マネジメントに関する書籍・論文はほとんどない。多くのプロジェクトマネージャは、試行錯誤でコミュニケーションをマネージしていると述べている。内田 [52]は、組織的なアセスメントを行っているにも関わらず損益悪化になっている失敗プロジェクトが複数あり、過去と同じ要因による失敗を繰り返していることが分かった。すなわち、過去の失敗が現在に十分に活かされていないという現状があると報告している。

村松 [53]の文献では次のように述べている。最近の経済環境において、システムインテグレーションに対する顧客ニーズが、短期化、低予算化を指向している。その結果、プロジェクトの数が増加する一方、一つのプロジェクトに配置できる要員の量を抑えざるを得ず、プロジェクト難易度が高くなってきている。また、PM の必要数が増加しており、相対的に経験が不足している PM も多くなってきている。このようにプロジェクトの難度があがるにもかかわらず、PM をはじめとする要員の経験が不足しており、プロジェクト

が失敗するケースが増加してきた。村松の会社では、プロジェクトの失敗を予防することを目的に、PMの個人の経験だけに依存するのではなく、組織全体の経験を有効に活用することでプロジェクトを成功へ導こうとしている。細谷ら[26]は、過去のプロジェクトの経験で得られている成功事例や失敗事例を社員で共有するためにハンドブックとして整備した。経験豊富な経営者層や上級プロジェクトマネージャのノウハウを収集し形式知化し、経験の浅いメンバへの配慮している。岡田 [51]は、開発対象システムの難易度が高くなるにつれて、システム開発プロジェクトの失敗も目立つようになってきている。この状況に対し、ユーザないしは各組織、ベンダー各社とも、失敗プロジェクトを分析して真の問題原因を探り出し、その分析の過程で得られたものを教訓としてまとめ、同種の失敗をしないようにその教訓を各組織、各プロジェクトに展開していると言っている。

岡田 [54]は、プロジェクトマネージャは多忙を極めており、蓄積された失敗ナレッジのデータベースがあっても、そこから積極的に有益なナレッジを探し出す余裕がない。失敗ナレッジをデータベースの中に蓄積し提供するだけでなく、業務のプロセスの中にビルドインし、プロジェクトマネージャが自ずと活用できるようにすることが重要であると述べている。

これらの文献から、プロジェクトのコミュニケーション・マネジメントに関わる書籍がほとんどないこと。一方で、プロジェクトの教訓を共有するためにまとめている会社があること。教訓をまとめても、参照する時間的余裕がないこと。などが分かる。本研究は、コミュニケーション・マネジメントに関する知見（教訓）を確認することを今後の課題としているが、これらの文献が示す状況はそのよりどころである。

## 2.6 関連研究から導くモデル化の根拠

本研究では ABS を使っている。シミュレーションモデルを作成する際に、実際のプロジェクトの全てのコミュニケーションプロセスを組み込むことが望ましい。しかし、プロジェクトのコミュニケーションパターンは多岐にわたり、複雑である。具体的には、コミュニケーションの相手とルート及び方向性、コミュニケーションの手段、コミュニケーションにおける情報伝達のパターン（情報の交換、交渉、調整）などである。それらのコミュニケーションをすべて組み込むことは、プログラム作成上難しいことだけではなく、プロジェクト全ての最大公約数的コミュニケーションパターンを採用する

観点からも望ましくない。従って、別途述べた Axelrod の KISS の原則に基づいて、モデルを単純化する。意味なくモデルを単純化することは、研究上好ましくない。そこで、関連研究からモデルの単純化の視点を下記のように導く。

#### 1) プロジェクトにおけるコミュニケーションの負荷

ソフトウェア開発プロジェクトでは、多くの時間をコミュニケーションに使う。プロジェクトの人数が多くなると、作業のほとんどがコミュニケーションに費やされる。

モデル化に当たって：本研究では、プロジェクト内の知識伝達の総合量を測定評価尺度とする。具体的には、「コミュニケーション効率」を測定する。

#### 2) コミュニケーションとプロジェクトの成功

部署を超えた協力のレベルが高いプロジェクトがより成功する。プロジェクトの成功のためには、チーム内のさまざまな知識を総合する必要がある。

モデル化に当たって：部署間のコミュニケーションをモデル化する。

#### 3) 専門領域を超えたコミュニケーション

プロジェクトが大規模になるほど、多くの専門家によってプロジェクトチームが構成され、互いの領域を超えたコミュニケーションを促す必要がある。

モデル化に当たって：部署を代表する PL の間の情報交換をモデル化する。

#### 4) Face to face コミュニケーションの重要性

プロジェクトにおいては、文書によるコミュニケーションでは十分でない。プロジェクトにおけるコミュニケーションは face to face が効果的である。

モデル化に当たって：接触コミュニケーションをモデル化する。具体的には、プロジェクトルーム空間の PL 同士の接触による情報交換をモデルに組み込み。文書もしくは e-mail によるコミュニケーションは、PM から PL への指示出しに限定する。

#### 5) 大規模プロジェクトのコミュニケーション

大規模プロジェクトでは、ひとつの問題を解決するのにも、複数の人間から関連情報を収集しなければならない。小規模プロジェクトで起こらない問題が大規模プロジェクトでは起こる。

モデル化に当たって：本研究では、シミュレーション結果について、大規模プロジェクトと小規模プロジェクトの比較を行う。

## 2.7 まとめ

はじめに、「プロジェクト」の定義を確認し、プロジェクト遂行に携わるプロジェクトチーム要員を定義した。また、プロジェクトの規模は、例外的なプロジェクトでなければ、プロジェクトの規模とそこに携わる要員数は比例すると考えて良いとし、プロジェクトの規模を定義した。

本研究は、連携性の高いソフトウェア開発プロジェクトを念頭に置き、連携性の高いプロジェクトほどプロジェクトチーム内部のコミュニケーション頻度が高く、コミュニケーション効率の良否がプロジェクト遂行効率を左右することを述べた。その上で、プロジェクトチームの組織は、プロジェクトの開始、進行、終結をたどる時間軸で、ゆるやかに変わってゆくが、本研究ではピーク時の組織をモデル化することを説明した。

本研究ではどのような因子が、コミュニケーション効率にどのような影響を及ぼすのかを調査し、コミュニケーション効率を高めるにはどうしたら良いのかについて調査することを問題設定とした。

問題設定を踏まえて関連研究を調査し、プロジェクトのコミュニケーションに関係する関連研究及びプロジェクトの **ABS** に関する過去の研究結果を分類した。

分類された関連研究からモデル化の根拠について述べた。プロジェクトの成功のためには、チーム内のさまざまな知識を総合する必要があるため、部署間のコミュニケーションをモデル化した。プロジェクトが大規模になるほど、多くの専門家によってプロジェクトチームが構成され、互いの領域を超えたコミュニケーションを促す必要があるため、部署を代表する **PL** 間の情報交換をモデル化した。プロジェクトにおいては、文書によるコミュニケーションでは十分でないことから、プロジェクトにおける接触コミュニケーションをモデル化した。大規模プロジェクトでは、小規模プロジェクトで起こらない問題が大規模プロジェクトでは起こることから、シミュレーション結果について、大規模プロジェクトと小規模プロジェクトの比較を行うこととした。

プロジェクトにおけるコミュニケーションはより複雑化してきており、プロ

ジェクトメンバ間の緊密な協力が必要であり相互依存関係を持っている。このような複雑な連携を演繹的な数学的手法でモデル化は難しいとし、**Agent-Based-Simulation** が有効であるとし、**KISS** 原理に従う必要があること述べた。

## 第 3 章 シミュレーションモデルの設計と実装

### 3.1 はじめに

現実プロジェクトのコミュニケーションからシミュレーションへのモデル化で、モデルに組み込むものとモデルに組み込まないものを明確にする。また、PL エージェントの知識レベル、PM から PL への指示メール出しの意味、PL 間の知識交換が何をモデル化しているかについて述べる。また、プロジェクト空間の PL の人口密度を一定にし、実験に無意味な影響が出ないようにしたことを説明する。最後に、研究で扱う評価尺度の定義と意味について示す。

### 3.2 現実プロジェクトのコミュニケーションとモデル

ソフトウェア開発プロジェクトは、多くの専門技術を組み合わせて、そのプロダクトであるソフトウェアをユーザに提供する。そのために、プロジェクトチーム内には各種の専門性をもった多くの専門家（PL・メンバ）が働いている。プロジェクトの目標を達成するために、全ての専門領域の仕事がプロジェクトの目指す方向に沿うよう、これらの専門家はプロジェクト全体の目標、課題、リスクなどの共通情報を共有する。同時に、自らの専門領域の仕事と関連する他の専門領域の仕事について、情報共有・調整（コーディネーション）するためにコミュニケーションを行う。

プロジェクトチーム内の専門家の仕事には、a)自らの専門領域の設計・製造・テストと、b)他の専門領域の専門家との調整（コミュニケーション）がある。b)のコミュニケーションは、プロジェクト内の専門家集団のリーダーである PL の重要な仕事である。専門領域間の調整内容はプロジェクトの進捗によって変わってゆく。この内容に関するコミュニケーションは、予定されたスケジュール表もしくは WBS に基づき、PM の指示により開始される。コミュニケーション効率を研究するという目的から、トラブル発生時の非定常なコミュニケーションはモデル化の対象外とする。

一般的なソフトウェア開発プロジェクトにおける、コミュニケーションのプロセス例を下記に示す。また、モデルに取り込んだかどうかも記す。



- PM は PL に対してプロジェクト全体の技術以外の一般情報をキックオフミーティング等で伝達する。(これは、一時で終了するので、効率を考慮する必要が少ないので、モデル化しない。)
- PL は PM の指示、もしくは PM から与えられたスケジュールに基づいて他の専門領域の PL と情報交換・交渉・調整を行い、自分の作業を行う。(この部分が効率的に行われる必要があり、モデル化の対象とする。) 一般的には、あらかじめミーティングに関わる情報を文書もしくは e-mail で交換しておく。
- 現状の作業内容と進捗、問題点を定例会議で、PL は PM に報告をする。(この部分は多くの時間を要しないためにモデル化しない。)
- 作業に掛れないタスクや問題があると、PL は PM にフィードバックを行う。(この部分は多くの時間を要しないためにモデル化しない。)
- 統括 PM は、PM のプロジェクト運営に問題がある場合、経営的な観点から、PM に指示を出す。(この部分はまれなのでモデル化しない。)
- まれに、メンバ (M) が、他の専門領域のメンバ (M) に技術上の問い合わせをすることがある。(この部分はまれなのでモデル化しない。)

コミュニケーション効率の研究を行う目的からモデル化を行っているので、次のようにした。実際のプロジェクトでは上記の他に、統括 PM と PM のコミュニケーションや、専門領域を超えたメンバ (M) 同士のコミュニケーションがある。一般的に、統括 PM はプロジェクトの実際の運営を PM に任せ、問題ある時にプロジェクト運営にかかわることから、効率と関わりないので本研究の対象から外すことにする。PL の下のプロジェクトメンバは第 5 章のアンケート調査の結果からも分かるように、コミュニケーションをする時間よりも、自らの専門業務を自己完結的に行う割合が高いことから、本研究の目的であるコミュニケーション効率の対象外なので、本研究のシミュレーションではモデル化しない。

シミュレーションモデルについては、実際のプロジェクトにおけるコミュニケーションを精緻にモデル化する方法もあるが、設定パラメータを設定する根拠もなく、結果が恣意的になる恐れがある。この研究では、プロジェクトのコミュニケーションの最大公約数的な部分をモデル化することにより、結論を一般化する。同時に、得られた結果をプロジェクトチームにおける実際のコミュニケーション状況を完璧に証明するためではなく、あくまで傾向を洞察するために限って用いるものであるという前提の下に、Axelrod[12]の提唱する KISS 原理に従って、プロジェクトにおけるコミュニケーション

を単純に設定した。

ABS は構造計画研究所が開発した artisoc [10]を利用して行った。シミュレーションモデルを単純化するにあたって、現実のプロジェクトにおけるコミュニケーションから組み込んだものと、組み込まなかったものを以下に説明する。

#### 1) モデルに組み込むもの

モデルに組み込む内容は下記のとおり。

- PM からの PL に対して、互いの専門情報の交換指示を出す。これはあらかじめ決められたスケジュールもしくは WBS に基づくが、モデルでは PM が持つテーマからランダムに送出される。シミュレーションとしては、特定の決められたスケジュールを設定するよりも、ランダムにした方が一般性を確保されると思われる。
- ランダムに埋まっているスケジュールの合間を縫って設定されるミーティングで、PL 間が互いの専門情報を交換する。これは、PL がプロジェクトルーム空間をランダムウォークして、接触することでスケジュール調整のランダム性をモデル化する。
- PL 相互のコミュニケーションの内、情報共有のための情報伝達をモデルに組み込む。実際の PL 同士のコミュニケーションには、互いの専門領域のインターフェースの調整やそれに伴う交渉がある。コミュニケーションの効率をモデル化する観点から、調整・交渉は、その結果を伝達することとモデル化する。
- PL 間でプロジェクトにおける専門固有情報を伝達することをモデル化する。一般的な専門知識を伝達することは、PL は他のプロジェクトで経験して知っていることが多いので除外しても差し支えない。
- プロジェクトルーム内の PL 人口密度一定化による PL 同士の接触確率の定率化を組み込む。これを組み込まないと、PL 数を増減させることによるプロジェクト規模変化のシミュレーションができない。

#### 2) モデルに組み込まないもの

モデルに組み込まないものは下記のとおり。

- PL から PM へのフィードバックは回数がまれで、効率が問題になることがないので外す。
- 統括 PM に関わるコミュニケーションは異常発生時などが多く、定常的でないのでモデルに組み込まない。一般メンバ間によるコミュニケーション

ョンはその比率が小さく、プロジェクトの方向を定める位置づけも少ないので除外する。

- PL 間の e-mail や文書による遠隔コミュニケーションは、接触コミュニケーション（ミーティング）の前に行われ、その一部と見なされるのでモデル化しない。
- コミュニケーションにおける相互調整・交渉は、その結果を伝達すると考えることにより、括り出してのモデル化はしない。
- 専門以外のプロジェクト全体情報の PM からの伝達は、頻度が少ないうえに、あまり効率の良し悪しが問題にならないのでモデル化しない。
- 一般的な専門外知識を初めから持っている PL がいるが、多くのプロジェクトで PL 間の情報交換の対象となるのはプロジェクト特有情報なので、一般的専門外知識を持っている PL をモデル化することはしない。
- 特定のスケジュールに基づいて、PM が順次情報交換テーマを選ぶことをモデル化することは、プロジェクトのモデルが一般化できないので行わない。
- 超大規模プロジェクトにおける PL の二層組織は、まれであるのでモデル化しない。

次節以降、上記に基づいて作成されたモデルの詳細について述べる。

### 3.3 シミュレーション空間におけるエージェントの配置

プロジェクトルーム空間は、エージェント（PL）の人口密度が一定になるように設定されており、PL 同士が出会う確率は、PL の数に関わらず一定である。人口密度とは、プロジェクトルーム空間の 1 セルあたりの PL 数で定義する。人口密度が高くと、ランダムウォークしている PL 同士が隣り合わせもしくは同一セルに繰る確率が高くなるため、コミュニケーション効率が良くなる。プロジェクトルーム空間は、その幅と高さで大きさが定義される。空間の幅とは、プロジェクトルーム空間の幅を何セルで構成するかであり、空間の高さとは、プロジェクトルーム空間の高さを何セルで構成するかを示すものである。それにより空間の面積が決まる。PL の数を空間面積で割ると人口密度（PL/セル）が出る。本研究では、シミュレーション結果に人口密度の影響が出ないようにするため、どのシミュレーションも人口密度がほぼ 1PL/4 セルとなるように設定している。

図 4 に示すように、エージェント（PM 及び PL）が動き回る「プロジェクト

ルーム」空間を二つのレイヤに分け、「知識交換レイヤ」及び「電子メール指示レイヤ」と名づける。ランダムに動き回る PM, PL エージェントを両レイヤに配置する。両レイヤの端点（壁）はループさせない。つまり、PM, PL の両エージェントはレイヤの端点（壁）にぶつくと来た方向に跳ね返る。両エージェントは、二つのレイヤで同一座標に位置する。電子メール指示レイヤでは、PM が PL に電子メールで情報交換の作業指示を出す。知識交換レイヤでは、PL が互いに隣接もしくは同一セル（ABS の空間を成す升目）に来るとコミュニケーションを行う。PM は、知識交換レイヤでは、PL とコミュニケーションは行わない。

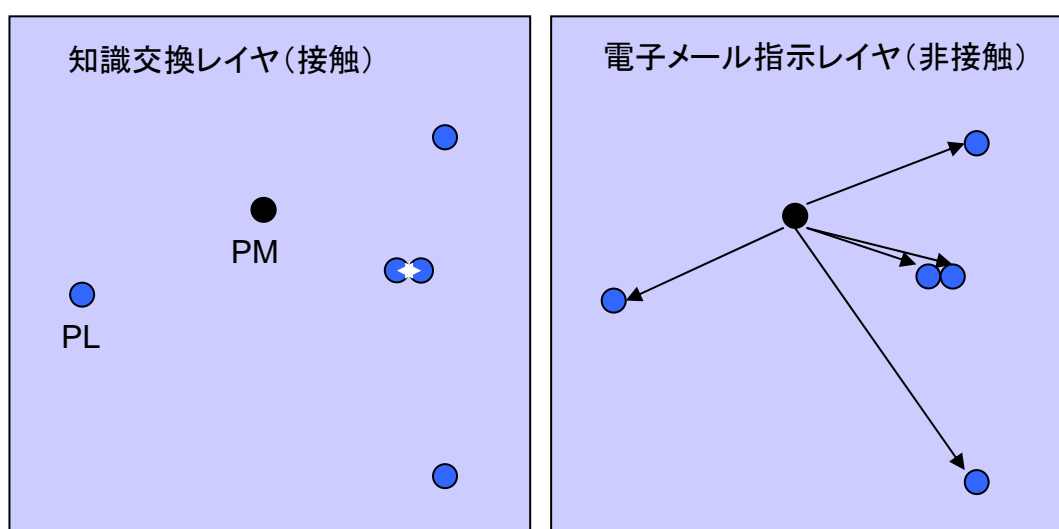


図4 PM・PL エージェントの存在する2レイヤのプロジェクトルーム空間

### 3.4 現実プロジェクト内での情報交換からシミュレーションモデルへ

一般的に実際のプロジェクトでは、プロジェクトに関する専門技術以外の情報（プロジェクトの目的、リスク、制約条件、ステークホルダ情報など）は、あらかじめ PM から各 PL に伝達されている。また、各 PL は同様なプロジェクトの遂行経験から、自分の専門技術領域外の技術知識を浅く広く持っている。担当するプロジェクトに特有な専門技術情報については、自らの専門領域に限って、あらかじめ仕様書等で知っている。一方、専門技術以外の領域の特有情報については、その専門 PL に聞くことによって入手される。つまり、プロジェクト特有な他領域の専門技術情報は、プロジェクト開始当初、一般的に PL は持っていない。

現実のプロジェクトでは、あらかじめ設定されたスケジュールもしくは WBS に基づいて、PM は PL に対して次にやるべき作業を指示する。もしくは、PL がスケジュールもしくは WBS 通りやるべきことを始めているかを PM が確認をする。本シミュレーションモデルでは、PM からの指示メールにより、PL 同士が情報交換を始める。

PL は自らの作業を遂行するために情報を集める。同時に、プロジェクトチーム内に情報を発信する。具体的には、関連する作業を担当する他の専門領域の PL からの情報を入手し、相手方の PL に対して情報提供を行う。例えば、サブシステム A の担当 PL はデータベース担当 PL と、技術の境界領域について設計・開発情報の交換を行う。シミュレーションモデルでは、情報交換をする PL 同士の知識レベルを比較し、高い知識レベルの方から低い知識レベルを持つ PL に情報が流れることにした。

各専門領域を担当する PL 間の情報交換は、まず、文書 (e-mail によるものも含む) でなされる。ソフトウェア開発プロジェクトでは、多くの場合、このような遠隔による情報交換では十分ではないために、PL 同士が直接ミーティングを持って、文書の行間を埋める。同時に、必要な設計上の調整を行う。PL 間のミーティングは、自分の作業や他のミーティングの間を縫って、時間調整のうえ設定されるか、たまたまプロジェクトルーム内で立ち話的に済まされる。シミュレーションモデルでは、ランダムウォークする PL 同士がたまたま接近すると情報交換を行う。これは、PL 同士が互いにスケジュールを調整してミーティングを行うことをモデル化するものである。

### 3.5 PL エージェントの知識レベル

PL は全員、担当する専門領域のプロフェッショナルである。現実のプロジェクトでは、各 PL はプロジェクト全般の情報は何らかの方法で知らされている。それ以外の他の専門領域の具体的な情報も持ち合わせている場合もある。シミュレーションモデルでは、単純化の為に自分の専門外の知識は、図 5 に示すとおり、最初は全く持っていないこととする。つまり、最高の知識レベルは 9 であるが、専門外については最初の知識レベルは 0 とする。逆に、専門領域の知識レベルは最高レベルの 9 である。

一人の PL がひとつの専門領域を受け持つという前提で、専門領域の数はプロジェクトルーム内の PL の数と同じとする。従って、図 5 において、PL の

持つ数字の桁数は PL の数と同じになる。

ここで、引用文献の内容を参照し、下記のようにシミュレーションモデルを設定した。

- 仲谷[6]は、プロジェクトのコミュニケーションにおいては、接触によるコミュニケーションが重要な意味を持つと述べていることから、エージェントの動き回る空間（プロジェクトルーム）を設定し、エージェントが重なり合うか、隣り合うセルにきた時にコミュニケーションすることとした。3つ以上の PL エージェントが重なり合うか、隣り合った場合でも、シミュレーションの一つのステップで全ての近隣 PL エージェントがコミュニケーションし合う。従って、二つ知識領域を代表する PL だけでなく、多くの PL が集うミーティング（合同ミーティング）も扱っていることになる。
- 実際のプロジェクトにおける PL 同士のコミュニケーションは、PM の指示を待たずに、スケジュールもしくは WBS に決められた内容について、自らミーティングを設定して情報交換を行うこともある。また、PL の自らの事情によりコーディネーションをしなければならないケースも想定される。このようなコミュニケーションを生起させる全ての状況をモデル化することは、モデルの複雑性そのものがシミュレーションの結果の影響を与える可能性がある（Axelrod[12]）ため、単純化して、電子メール指示が PM からあることで、PL 間のコミュニケーションを開始させる。

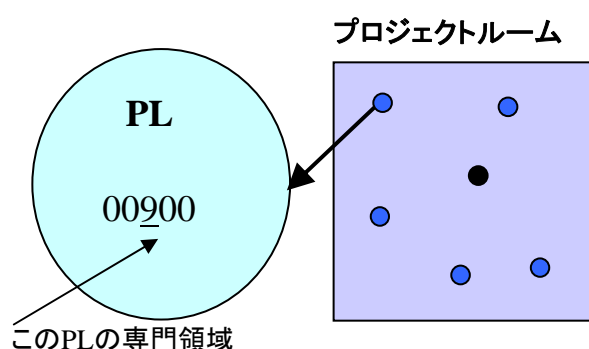


図5 PL エージェントの知識レベル

### 3.6 PM から PL への指示メール出しの概念

PM が電子メールで、PL に対して、コミュニケーションすべき専門領域を指

定する。このコミュニケーション指示は、用意された領域についてランダムに PL に対して送出される。単純化のために、コミュニケーション指示は、各専門領域に関わるテーマに紐づいた 9 種類の指示内容とし、それらは専門領域に関してランダムに送出される。

シミュレーションの開始前状態として、5 つの異なる専門領域のテーマを各々 9 つ持っている状態を 99999 と表す。5 つの専門領域となる理由は、図 5 で PL の数が 5 人だからである。ここで、PM の数字は、PL の持つ知識のレベルを表す数字と異なり、知識交換すべきテーマの残数を表す。このコミュニケーション（知識交換）すべきテーマをランダムに順次 PL に指示するに従って、各桁の数字が少なくなり、ついには 00000 となる。（図 6） PM がプロジェクト遂行に必要な全てのテーマのコミュニケーション指示を出した時点で、プロジェクトは完了したものとする。PM の数字の各桁で 0 は指示すべきテーマが無くなったことを示す。

これを単純化したモデルで表すために、下記を前提とした。

- Turner[17]が言うプロジェクトにおけるコミュニケーション能力から導かれる分類（情報交換・調整・交渉）をそれぞれモデル化するのではなく、単純化し、PL が「知識」を交換するという動きでコミュニケーションを代表させた。つまり、このモデルでの PL 間のコミュニケーションは、「知識」という単位の単なる情報・調整結果・交渉結果を伝播させることとする。
- 実際のプロジェクトにおける PL は、自らが専門とする領域以外の知識や情報も持ち合わせていることが多い。このシミュレーションにおける「知識」は、その専門の一般的知識ではなく、そのプロジェクト特有の情報と位置づける。この想定を元に、自分の専門領域以外の「知識」を全く無いと単純化した。
- 実際のプロジェクトでは、専門「知識」以外にも、プロジェクト全体に関わる情報を共有しなければならない場合が多い。これらの情報は、接触によるコミュニケーション以外のプロジェクト全員参加のミーティングにて伝達されるとの想定で、モデル化からはずした。

既往論文の調査結果で示した通り、PM はあらかじめ設定されたスケジュールもしくは WBS から、その時点で必要な作業を各 PL に伝える。その際に、このプロジェクト特有の専門技術的な制約条件・リスク条件を示し、その条件に基づいて互いに関連する専門領域の PL 間で、設計・開発上の調整を行

わせる。

例えば、PM から各 PL に、旧システムから新システムへの移行条件や、新システムが運用開始時にうまく動かなかった場合のリスクが伝えられる。加えて、それらに対する対応方針などの情報も PM から各 PL に伝えられ、それに基づいて専門領域担当 PL 間で設計・開発上のすり合わせが行われるなどである。

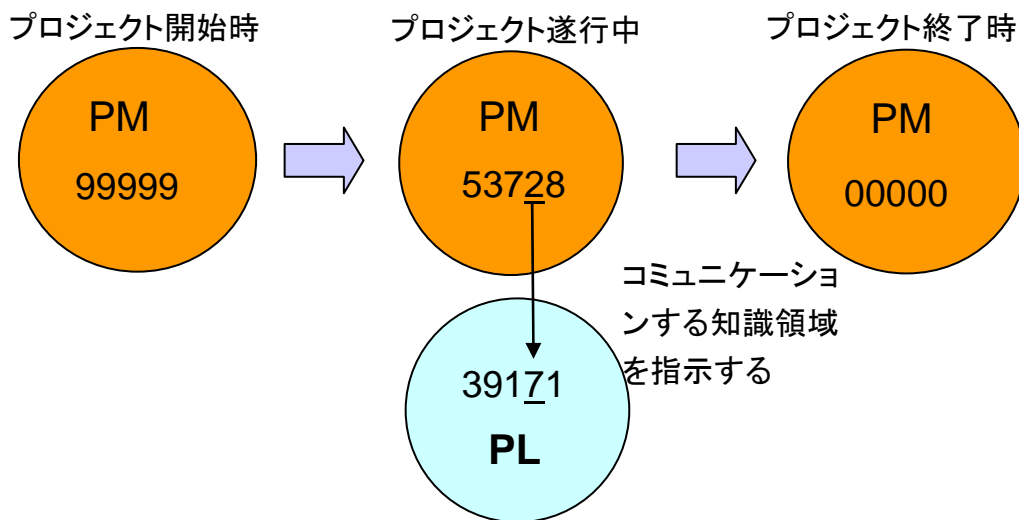


図6 PM から PL への指示メール出しの概念

### 3.7 PL 間の知識交換

PL はプロジェクトルーム内をランダムに動き回る。PL は他の PL が同一もしくは隣のセル (ABS では空間をセルに区切っている) に来ると、両 PL はシミュレーションの 1 ステップ (コンピュータシミュレーションを行う際に決められた手順を一回たどる単位で、単位時間と考えてもよい。モデルでは複数のステップだけ同じ場所に留まる設定が可能。) だけ同じセルに留まり、PM から受領した作業指示に関する「知識」について、PL は持てる「知識」を、それを持っていない PL に与える。PL は、PM から作業指示された専門領域のテーマに関わる「知識」交換のみを行う。

図7 は、PL エージェント同士がどのようにコミュニケーションするかを示すものである。ここでは、理解しやすいように、「知識交換レイヤ」及び「電子メール指示レイヤ」を同一面上に表現している。図7では、PM が左から



3番目の専門領域について知識交換するように PL に指示を出す。それを受け取った PL の内、他の PL と接触（隣のセルか同じセルに居ること）している PL がお互いに知識交換を行う。この図では、左上の PL（知識 41796）が右下の PL（知識 29381）に対して、左から三番目の知識領域の知識（数字の下にアンダーラインをしているもの）をひとつ与えている。これによって、この PL は「知識」が増え、「知識」組み合わせが 29381 から 29481 に変化する。この「知識」を与えるコミュニケーションは、「知識」が高い（数字が大きい）方から、小さい方に対して一方通行で行われる。「知識」のレベルが同じ場合は、コミュニケーションは行われず、すぐに2セル離れる設定としている。PM からコミュニケーションする指示が無い「知識」に関して、互いの持てる「知識」が異なっても、それについては、知識交換は行わない。

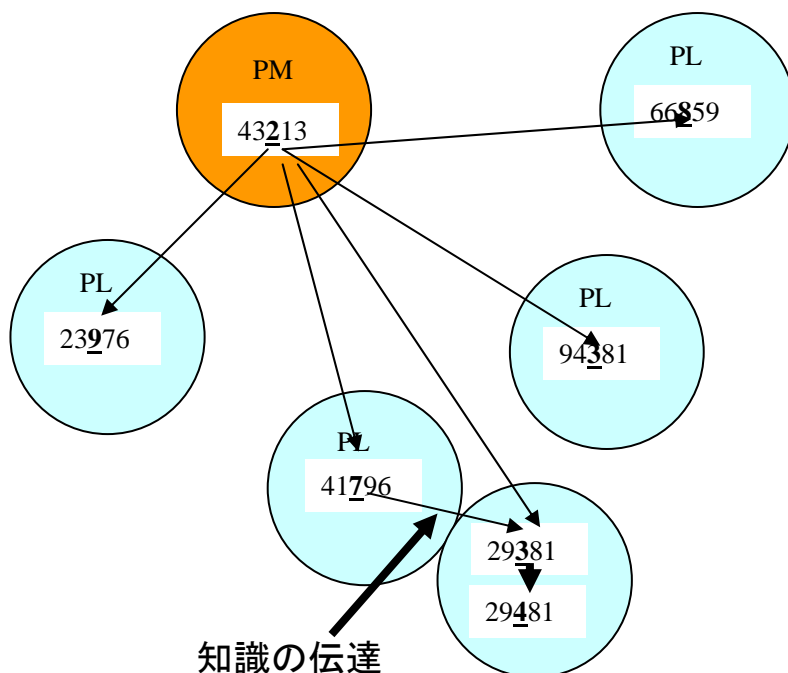


図7 PM からの指示に基づく PL 間の知識交換

実際のプロジェクトでは、PM から作業指示があり、PL 同士のミーティングによる対面コミュニケーションが必要な場合でも、PL 同士の時間的都合によって、すぐにはミーティングが開催されず、互いの都合を調整してミーティング時間を設定する。短時間で済む情報交換は、たまたま出会った際に、立ち話ですませる。これを本研究のシミュレーションモデルでは PL のランダムウォークに代えている。

特定領域の情報も一部の他領域 PL に必要なものと、ほとんどすべての他領域 PL に必要な情報がある。後者の例として、システムがクラスター化されている場合、インフラ設計、データベース設計、サブシステム設計に影響する例がある。前者と後者、両方の情報を差別してモデル化すると設定条件が複雑化し、シミュレーション結果を恣意的にする可能性があるためにモデルでは差別しない。

PL 同士が行う技術情報交換及び設計・開発上のすり合わせのミーティングは、通常、その時点で必要なものに限られて行われる。スケジュール上かなり先に必要なものについては、それまでの設計・開発によって左右されるので、基本的なもの以外、あらかじめミーティングの対象となることはない。したがって、モデルでは、PM から PL に指示が出た知識の情報交換に限ってしまっても、現実と大きく相違することはない。

実際のプロジェクトでは、専門領域によって、他領域とコミュニケーションを多く必要とする領域と、そうでない領域がある。ビル建設のプロジェクトに例えてみる。ビルの躯体に関わる専門領域の情報は、ビルの基礎を作る土木領域や、電気領域、上下水道領域に関わる。一方、ビルの外構に関わる領域の情報は他の専門領域であまり必要とされない。この状況を区別してモデルに組み込むことは、モデルの単純化を妨げる (Axelrod[12]) ために、全ての知識領域のコミュニケーションすべきテーマを単純に 1 から 9 の 9 種類とした。

作業指示は、一般のプロジェクトでは、スケジュールもしくは WBS に基づいて PM が促すものであるので、PM から一方的に PL に指示を出す単純化を行った。現実のプロジェクトでは、スケジュール遅れやその他の理由により、PL が PM と交渉して、作業順序を変えることがある。しかし、それは例外的なものなので、PL が PM と作業指示について交渉するようなモデル化はしていない。

実際のプロジェクトでは、PL 同士は電子メールや文書によるコミュニケーションも行う。ここでは、岡村[1]、仲谷[6]、Pinto[19]、Michalski[16]らが有効であり、必要であるとする face-to-face コミュニケーションのみを扱う単純化を行う。

現実のプロジェクトでは、多くのコミュニケーションが、会議をあらかじめ計画的に設定して行われる。関連研究から、計画的でない（ランダムに発生する）会話や距離的に近い人によるコミュニケーションが有効であることが分かる。（仲谷[6]）このシミュレーションモデルでは、ランダムなコミュニケーションのみを組み込むこととする。この前提に基づき、PL がランダムに動き回り、接触による「知識」交換をする。会議をあらかじめ設定するとしても、既にランダムに決まっている別の会議スケジュールを避けてこの会議が設定されるとし、ランダムに出会ってコミュニケーションを行うことと同様とすることによって単純化する。

実際のミーティングでは、PM から指示の無いコミュニケーションテーマがあった場合、その場で話し合われる場合もまれにある。また、PM から指示のあった専門領域について複数のテーマのコミュニケーションが行われる。このシミュレーションでは、PM から指示のあったコミュニケーションテーマについてのみ話し合われるものと単純化する。

### 3.8 PM と PL の持つ数字の意味

PL は各知識領域の専門家と位置づけ、その専門数だけ知識を表す数字の桁を設定する。これは、PM と PL の持つ数字の桁を規定する。数字の桁数は PM と PL と同じであるが、意味が異なる。

PL の数字は各専門領域の知識レベルを表し、大きな数字ほど高い「知識」を持つことを意味する。PL の数字はプロジェクトが進行する（シミュレーションが進む）につれて大きくなる。PM の数字は各桁の示す専門領域で PL が交換すべき「知識」テーマの残数を示し、PM が PL に知識交換（コミュニケーション）を指示することにより、この数字（コミュニケーションすべきテーマの残り数）は減ってゆく。

PM からコミュニケーションする指示が無い「知識」に関して、PL は互いの持てる「知識」レベルが異なっても、それについては、知識交換は行わない。PL 間で交換される「知識」は、各 PL の専門領域に関わる 9 つ（1～9）の内容とする。PL の数字 0 は、交換すべき「知識」が無いことを示す。

### 3.9 研究で扱う評価尺度

プロジェクトにおけるコミュニケーションの既存研究はあまり成果が上がないと言われている。その理由の一つは、コミュニケーションの有効性に関して、測定する評価尺度が明確ではなかったことである。(込山 [45]) 本論文では、研究に際して「コミュニケーション効率」という評価尺度を新たに導入した。

この研究では、PL 一人あたりのコミュニケーションの効率を評価尺度として扱う。

コミュニケーション効率の定義は：

$$\eta = \Delta K / (n \times s) = \Delta K / (n \times 9n) = \Delta K / (9n^2) \quad \text{式 (1)}$$

ここで、

$\eta$  : コミュニケーション効率

$\Delta K$  : プロジェクトチーム内に拡散した知識数

$s$  : プロジェクトに費やした時間 (ステップ数)

$n$  : プロジェクトリーダー (PL) の人数

つまり、「プロジェクトチーム内で、単位時間に、PL 一人当たり、どれくらいの知識が拡散したか。」とする。

図 8 に示すように、シミュレーション前のプロジェクトチーム内の知識数の合計  $K$  は、元から PL が持つ知識数の合計となる。それが、図 9 に示すように、プロジェクト終了時のプロジェクトチーム内知識数の合計  $K'$  は知識が伝搬した分だけ増える。この増分を  $\Delta K$  とする。一方、プロジェクト終了までのステップ数  $s$  は、モデルの定義から図 10 に示す通り  $9n$  である。

下記の式が成り立つ。

$$0/(9n^2) = 0 \leq \eta \leq 9n(n-1)/(9n^2) = 1-1/n \quad \text{式 (2)}$$

これは、シミュレーション効率の最大・最小値の取りうる範囲を示すものである。コミュニケーション効率の取りうる最小値は 0 である。最大値は、 $n$  が大きいほど小さくなることが分かる。 $n=2$  の場合は、 $\eta = 0.5$  となるが、 $n = \infty$  の場合は、1 となる。

シミュレーション開始前

PL <sub>1</sub> : <u>9</u> 0000	k <sub>1</sub> =9	} K=9n=9x5
PL <sub>2</sub> :0 <u>9</u> 000	k <sub>2</sub> =9	
PL <sub>3</sub> :00 <u>9</u> 00	k <sub>3</sub> =9	
PL <sub>4</sub> :000 <u>9</u> 0	k <sub>4</sub> =9	
PL <sub>5</sub> :0000 <u>9</u>	k <sub>5</sub> =9	

図 8 シミュレーション前の PL の知識数

シミュレーション終了後

PL <sub>1</sub> : <u>9</u> 3526	k <sub>1</sub> =25	} K'=K+ΔK =9n+ΔK =9x5+99
PL <sub>2</sub> :4 <u>9</u> 167	k <sub>2</sub> =27	
PL <sub>3</sub> :72 <u>9</u> 38	k <sub>3</sub> =29	
PL <sub>4</sub> :846 <u>9</u> 5	k <sub>4</sub> =32	
PL <sub>5</sub> :3874 <u>9</u>	k <sub>5</sub> =31	

図 9 シミュレーション後の PL の知識数

s=0	PM:99999
s=1	PM:99899
s=2	PM:89899
s=3	PM:89799
s=4	PM:89789
s=5	PM:79789
.....	
.....	
s=9n	PM:00000 <b>プロジェクト終了</b>

図 10 プロジェクト終了までのステップ数

PL は交換した「知識」で、プロジェクトの仕事をこなす。プロジェクトが

終了した段階で、プロジェクト内全ての PL が得た「知識」(情報の総量)が豊富であれば、プロジェクトの完成度は高くなると考える。反対に PL が得た「知識」(情報の総量)が少なければ、互いに関連する専門領域の PL 間の調整が未熟で、プロジェクトによって提供される製品やサービスの内容と質に問題があると言える。ステップあたり、PL ひとりあたりの「知識」増分が大きいほど、コミュニケーション効率が良いと考える。

専門的な技術情報についてすべての PL が知っておかなければならないものがある。例えば、プロジェクトで達成しなければならないサービスレベルアグリーメントの内容は、その例である。サービスレベルアグリーメントを達成するために行う各専門領域 PL が行う設計・開発の詳細は、プロジェクト内部のすべての PL が共有することが望ましく、このような情報のプロジェクト内部への伝達はプロジェクトの成功率に大きく影響する。

### 3.10 まとめ

プロジェクトのコミュニケーション効率の研究を行う目的からモデル化を行っているので、PM と PL のコミュニケーションをモデル化した。エージェントが動き回る「プロジェクトルーム」空間を二つのレイヤに分け、ランダムに動き回る PM, PL エージェントを両レイヤに配置する。電子メール指示レイヤでは、PM が PL に電子メールで情報交換の作業指示を出す。知識交換レイヤでは、PL が互いに隣接もしくは同一セルに来るとコミュニケーションを行う。PM からの指示メールにより、PL 同士が情報交換を始める。高い知識レベルの方から低い知識レベルを持つ PL に情報が流れる。

PL は全員、担当する専門領域のプロフェッショナルである。シミュレーションモデルでは、単純化の為に自分の専門外の知識は、最初は全く持っていないこととする。つまり、専門外については最初の知識レベルは 0 とする。逆に、専門領域の知識レベルは最高レベルの 9 である。

プロジェクトにおけるコミュニケーションの既存研究はあまり成果が上がっていないと言われている。その理由の一つは、コミュニケーションの有効性に関して、測定する評価尺度が明確ではなかったことである。本論文では、研究に際して「コミュニケーション効率」という評価尺度を新たに導入した。「プロジェクトチーム内で、単位時間に、PL 一人当たり、どれくらいの知識が拡散したか。」とする。コミュニケーション効率は、0 から  $1 - 1/(PL$

の人数)の間の値を取る。

コミュニケーション効率が高ければ、プロジェクトの完成度は高くなると考える。反対に低ければ、互いに関連する専門領域の PL 間の調整が未熟で、プロジェクトによって提供される製品やサービスの内容と質に問題があると言える。

## 第4章 シミュレーションの実施と考察

### 4.1 はじめに

本研究で行われたシミュレーションのフロー、シミュレーションで設定するパラメータ（PLの数、指示メールの送出サイクル）及びABSを実際に実行した画面について説明する。シミュレーションから得られた結果から、PLの数の影響、指示メールの送出サイクルの影響、効率の良いコミュニケーション方法（知識交換テーマを遡る効果、PMの指示サイクルとテーマの遡及の交互作用、ミーティング時間とテーマの遡及の交互作用）について考察する。

### 4.2 PLの数と指示メールの送出周期の影響

PLの数とPMからPLに出される指示メールの送出周期の影響を確認する。

#### 4.2.1 シミュレーションのフロー

本シミュレーションで設定するパラメータについて説明する。

- PLの数：プロジェクトチーム内のPLの人数を示す。この数が大きければ大規模プロジェクト、小さければ小規模プロジェクトを意味する。
- 指示メールの送出サイクル：PLが相互に知識交換を行うきっかけを作る指示メールを、PMが出す周期を意味する。この周期が大きいことは、PMからの指示メールの発信がゆっくりであることを示す。
- 乱数シード値：シミュレーションソフトウェアで使う乱数の初期値を定義する。これは、実験のランダム性と再現性を担保する。乱数シード値をシミュレーションモデルに入力することにより、同じシミュレーションを再現することが可能である。

シミュレーションは、表1に示したパラメータの全ての組み合わせで行われた。「PLの数」、「指示メールの送出サイクル」を変えた。同一の「PLの数」、「指示メールの送出サイクル」組み合わせで、乱数シード値を10回変えてシミュレーションを行い、その平均値をグラフ化した。



表 1 PL の数と指示メールの送出サイクル影響実験のパラメータの組み合わせ

パラメータ名	パラメータの値
PL の数 n	4, 8, 12, 16, 20
テーマの遡及範囲	1
指示メールの送出周期	1, 2, 3, 4, 5
留まるステップ数（ミーティング時間）	1

図 11 は、一つのシミュレーションで行われる PL 同士の知識交換フローを示す。

PL はプロジェクトルーム（接触空間）に条件で設定された人数ランダムに配置される。最初のシミュレーションステップで、PL はランダムな方向に 5 ステップ直進する。

プロジェクトの進捗に従って、PL 同士が情報交換を必要とするテーマについて、PM から PL に指示が出る。

PL は「知識交換レイヤ」をランダムに動き回っている。ランダムに動き回った結果、同一もしくは隣のセルに、ひとつもしくは複数の PL が来ると、一つのシミュレーションステップの間、そのセルに留まり、コミュニケーションを行う。これは、時間的にランダムに設定した会議に出席するか、偶然に出会った機会を捉えて「知識」提供の機会とすることを意味している。

コミュニケーションは、交換する「知識」（PM からの指示があった専門領域のテーマについての情報）があるかどうかを確認しあうことから始まる。二人の PL の出会いであれば互いの、二人を越える PL の出会いであれば全員の持っている「知識」に違いがあるかどうかを確認する。違いがあれば、これらの PL はひとつのシミュレーションステップの間そのセルに留まり、知識交換を行う。相手の持っていない「知識」があれば、ひとつのテーマだけについて「知識」を相手に提供する。自分の持っていない「知識」があれば、ひとつのテーマだけについて「知識」をもらう。

PL 全員でこのコミュニケーションを行う。互いの持っている「知識」に違いが無ければ、それらの PL はそのセルに留まることなく、そこを離れて再

びランダムに動き出す。また、ひとつのシミュレーションステップで「知識」を交換した後も同様にランダムに動き出す。

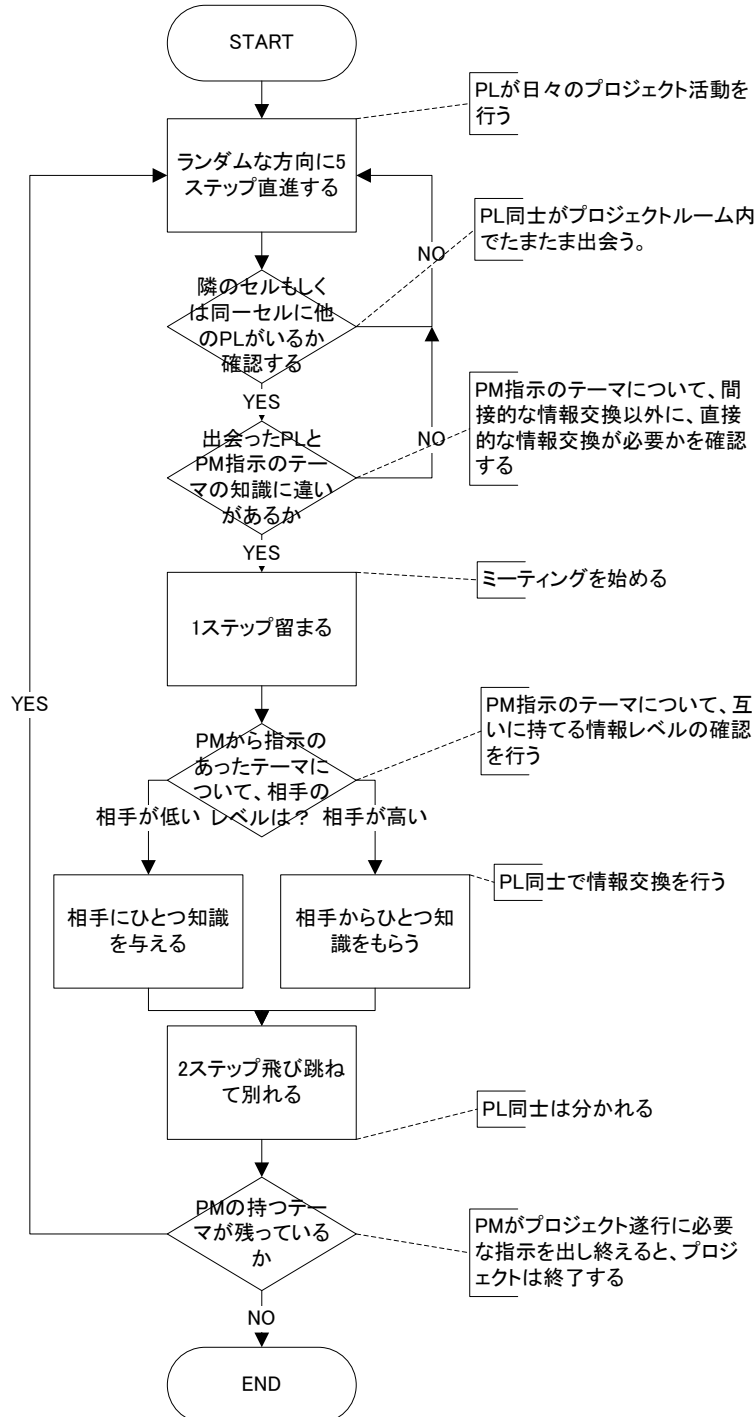


図 11 PL の数とメールの指示メールの送付サイクルのシミュレーションプログラムのフローチャート

PM が持つテーマが全て払いだされていなければ、再びシミュレーションフローの最初に戻る。PM の持つ全てのテーマが払いだされている場合は、プロジェクト終了として、シミュレーションを終了する。

このプログラムの実行で注意しなければならないのは、「0 接触空間」の PM0 と PL0, 「1 メール空間」の PM1 と PL1 の各プログラムの実行順序である。これを設定する画面が、シミュレーションソフトウェア artisoc 上のメニュー画面の「設定」にある。その順を誤るとエラーで落ちるか、再現性が確保されなくなる。

#### 4.2.2 シミュレーション結果と考察：PL の数の影響

PL の数  $n$  の影響を図 12 に示す。横軸は PL の数、縦軸はメールの送付周期を変化させた平均値としてのコミュニケーション効率を示す。グラフの各線は、シミュレーションソフトウェアを動かす時の乱数シード値を変えたものである。

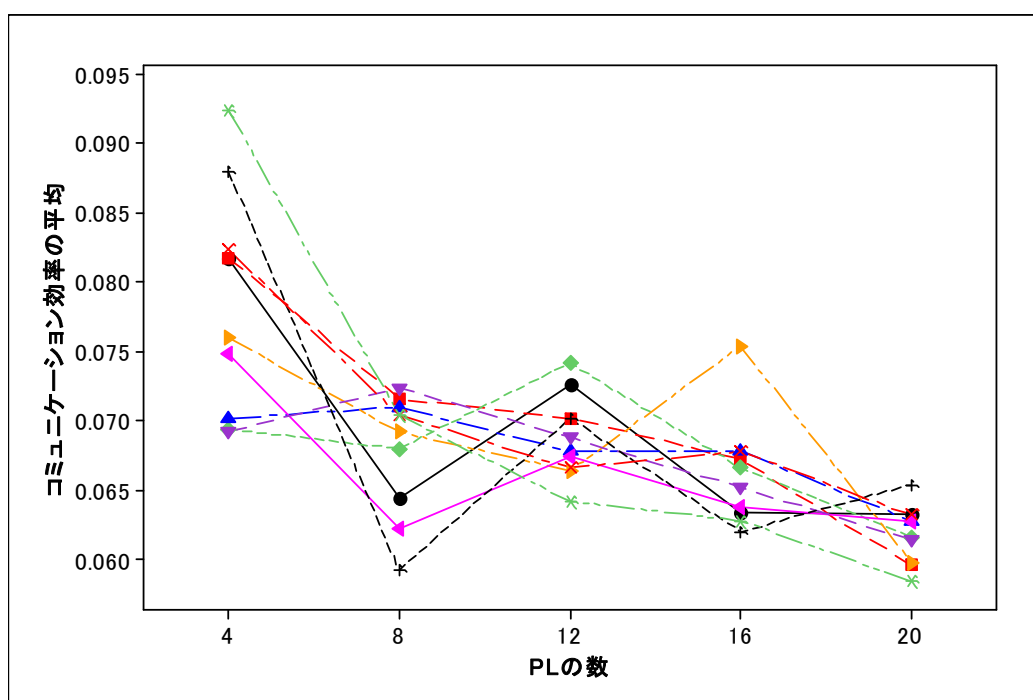


図 12 PL の数とコミュニケーション効率の関係

PL の数  $n$  が増えると、コミュニケーション効率は減少する。つまり、プロジェクトの規模が大きくなるほどコミュニケーション効率が悪くなる。コミ

コミュニケーション効率が減少するという事は、シミュレーション終了時点での、プロジェクトチーム内の知識の拡散総量が小さいことを意味する。

この結果は、

$$\eta \leq 9n(n-1)/(9n^2) = 1-1/n \text{ 式 (2)}$$

から、コミュニケーション効率  $\eta$  は PL の数  $n$  が大きくなるほど取りうる値が大きくなるが、この図 12 のグラフから；

$$\eta = \Delta K / (n \times s) = \Delta K / (n \times 9n) = \Delta K / (9n^2) \text{ 式 (1)}$$

の分母に入っている  $n^2$  が、より式 (2) の  $n$  より効く可能性がある。このようなシミュレーション結果が出る中、如何に式 (1) の  $\Delta K$  を大きくするかが、事節以降の研究である。

#### 4.2.3 シミュレーション結果と考察：指示メールの送出サイクルの影響

図 13 は、横軸に指示メールの送出サイクルを取り、縦軸にコミュニケーション効率の平均を示す。

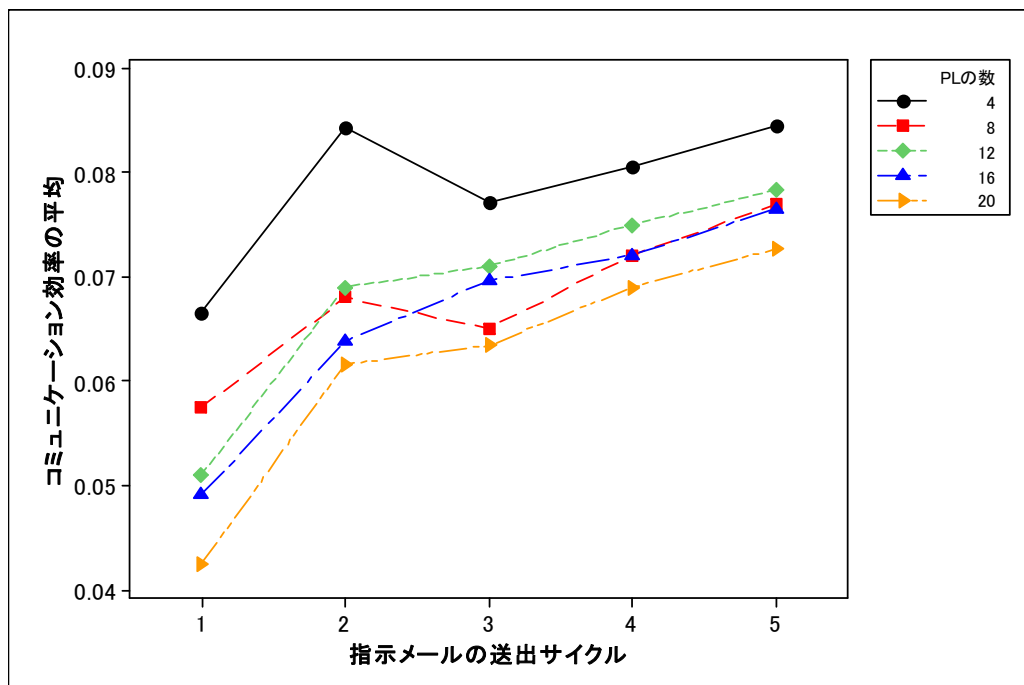


図 13 コミュニケーション効率の交互作用

PM からの指示メールの送出サイクルを長くするほど、コミュニケーション

効率は高くなる。但し、指示メールの送出サイクルが大きくなるとともに、コミュニケーション効率は寝てくることから、指示メールの送出サイクルを大きくすることは、それを大きくし過ぎても意味がないことが分かる。

PM からの指示出しをゆっくりすると、コミュニケーション効率は上がる。ただし、それはプロジェクトの納期を延ばすことになる。PM は、一定のプロジェクト納期の中で、適切な程度にゆっくりと指示を出すと良い。しかし、指示メールの送出サイクルがあまり大きくなるとその効果が薄まることを考えておく必要がある。また、本シミュレーションのパラメータを実際のプロジェクトのパラメータにそのまま置き換えられるわけではない。図 13 でグラフが寝てくる指示メールの送出サイクル=2 の値が、実際のプロジェクトに於いてどの程度になるかは、今後の研究の結果を待たなければならない。

### 4.3 効率の良いコミュニケーション方法

テーマの遡及範囲（ミーティングの議題数）とミーティング時間の影響についてシミュレーションした結果と考察を述べる。

#### 4.3.1 シミュレーションのフロー

シミュレーションモデルでは、「テーマの遡及範囲」として、指定ステップ数だけ遡った（以前に PM が出した）テーマについても情報の伝達が可能な機能を使用している。これは、ミーティングの議題を増やすことを模している。

また、ミーティング時間を長くする設定が可能にするために、PL エージェントが接触し、その場に留まるステップ数を設定数だけ延長し、情報交換を継続する機能を持たせている。乱数シード値を変えた 10 回のシミュレーションを行っている。結果は 10 回の平均値をグラフ化する。

シミュレーションのフローを図 14 に示す。影を付けたプロセスがこのシミュレーションで新たに使う機能である。

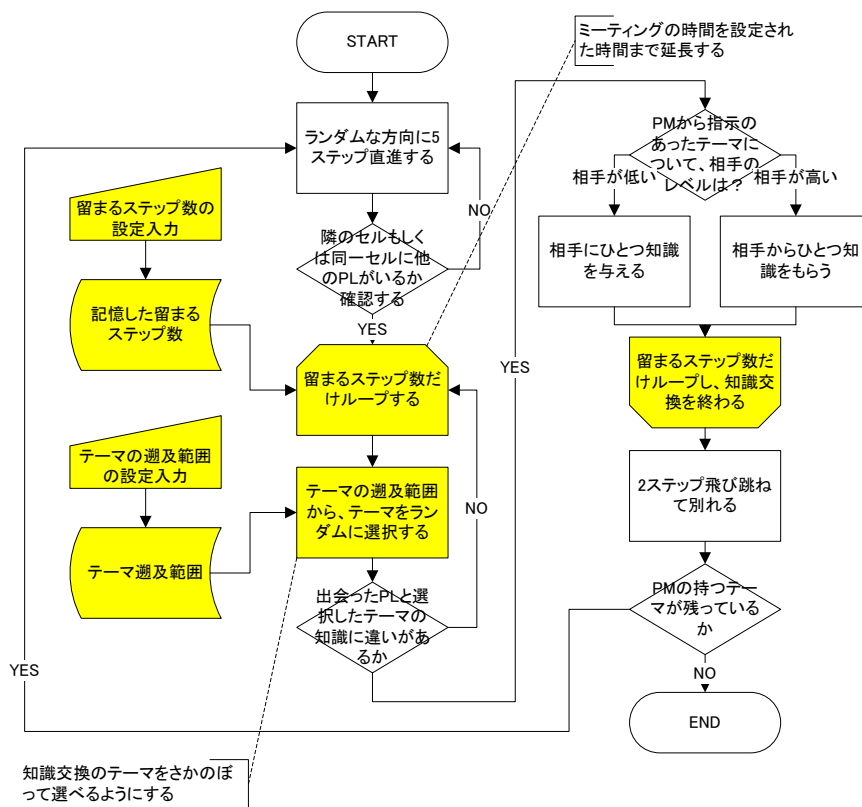


図 14 効率の良いコミュニケーション調査のためのシミュレーションプログラムのフローチャート

PL はプロジェクトルーム（接触空間）に条件で設定された人数ランダムに配置される。最初のシミュレーションステップで、PL はランダムな方向に 5 ステップ直進する。

プロジェクトの進捗に従って、PL 同士が情報交換を必要とするテーマについて、PM から PL に指示が出る。

PL は「知識交換レイヤ」をランダムに動き回っている。ランダムに動き回った結果、同一もしくは隣のセルに、ひとつもしくは複数の PL が来ると、あらかじめ入力されたステップ数（留まるステップ数）の間、そのセルに留まり、コミュニケーションを行う。これは、時間的にランダムに設定した会議に出席するか、偶然に出会った機会を捉えて「知識」提供の機会とすることを意味している。留まるステップ数は PL 同士が行うミーティングの長さを決める。

コミュニケーションは、交換する「知識」（PM からの指示があった専門領

域のテーマ)があるかどうかを確認しあうことから始まる。二人の PL の出会いであれば互いの、二人を越える PL の出会いであれば全員の持っている「知識」に違いがあるかどうかを確認する。違いがあれば、これらの PL は設定されたシミュレーションステップの間そのセルに留まり、知識交換を行う。相手の持っていない「知識」があれば、設定されたテーマ数について「知識」を相手に提供する。自分の持っていない「知識」があれば「知識」をもらう。この時に、PM から指示のあった専門領域のテーマを遡って「知識」交換の対象とできるように、あらかじめ条件設定に決められたテーマを遡及して扱うことが可能なようにプログラムが組まれている。

PL 全員でこのコミュニケーションを行う。互いの持っている「知識」に違いが無ければ、それらの PL はそのセルに留まることなく、そこを離れて再びランダムに動き出す。また、ひとつのシミュレーションステップで「知識」を交換した後も同様にランダムに動き出す。

PM が持つテーマが全て払い出されていなければ、再びシミュレーションフローの最初に戻る。PM の持つ全てのテーマが払い出されている場合は、プロジェクト終了として、シミュレーションを終了する。

このプログラムの実行で注意しなければならないのは、「0 接触空間」の PM0 と PL0, 「1 メール空間」の PM1 と PL1 の各プログラムの実行順序である。これを設定する画面が、シミュレーションソフトウェア *artisoc* 上のメニュー画面の「設定」にある。その順を誤るとエラーで落ちるか、再現性が確保されなくなる。

図 15 は、ABS を実際に実行した画面である。右側(A)が「1 メール空間」(非接触空間)で、PM が PL に対してコミュニケーションするテーマを電子メール送信する状況が表示されている。左側(B)が「0 接触空間」(接触空間)で、赤い丸が PM を表し、数字のみのものが PL を表している。右上(C)に「PL の数」、「記憶遡及範囲」、「メールの送出周期」、「留まるステップ数」を設定するコントロールパネルが表示されている。最も左(D)に表示されているものは、ABS の空間、エージェント、及びそれに属する変数をツリー構造で示したものである。最下段(E)が「コンソール画面」で、シミュレーション結果を表示する場所である。

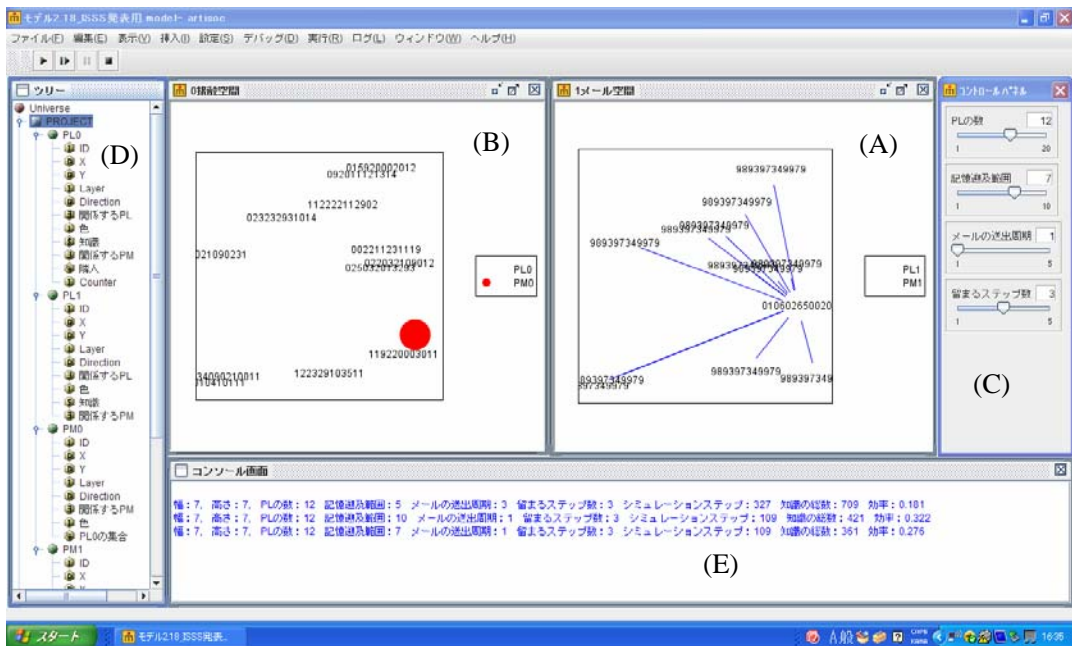


図 15 実際のシミュレーション画面の例

#### 4.3.2 知識交換テーマを遡る効果

ミーティングにおいて、知識交換テーマを遡る効果を見るために、表 2 の条件でシミュレーション実験を行った。「テーマの遡及範囲」（ミーティングの議題数に当たる）とは、PM が出した情報交換テーマを遡って PL が知識交換をできるように設定するもので、設定数値分前の PM 指示テーマについても PL 同士の知識交換が可能になる。

表 2 PL の数と知識交換テーマを遡る効果の交互作用実験パラメータの組み合わせ

パラメータ名	パラメータの値
PL の数 n	4, 8, 12, 16, 20
テーマの遡及範囲	1, 3, 5, 7, 9
指示メールの送出サイクル	3
留まるステップ数（ミーティング時間）	3

シミュレーション結果を図 16 に記す。横軸にテーマの遡及範囲，縦軸にコミュニケーション効率を取っている。シミュレーションは，PL の数を変えて 5 本の線にしている。ここで，指示メールの送出サイクル：3 ステップ。



ミーティング時間：3 ステップ。グラフの線はいずれも乱数シード値 10 回の平均値である。

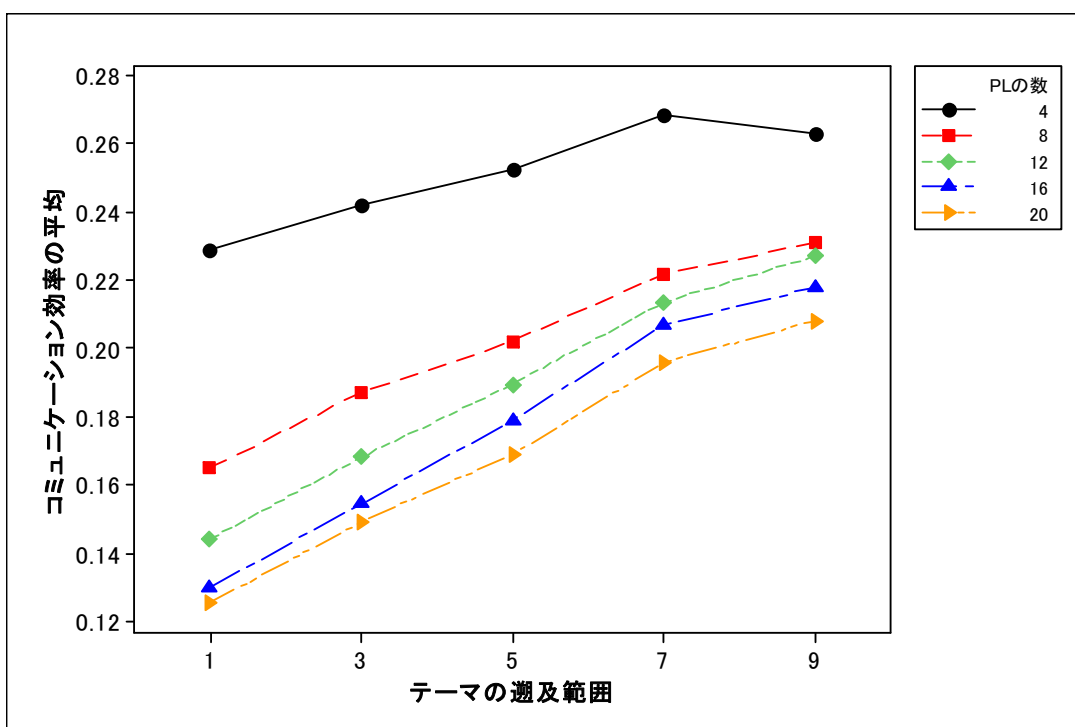


図 16 PL の数から見たテーマの遡及範囲とコミュニケーション効率の関係

#### 4.3.3 シミュレーション結果と考察：知識交換テーマを遡る効果

テーマの遡及範囲（議題の数）を多く取ると，PL の数に関わらずコミュニケーション効率が上がる．また，少しであるが PL の数が大きいほど，グラフの線の傾きが大きい．しかし，PL の数  $n$  が 4 の時に，グラフが寝てくることが伺え，その効果も限度があると言える．PL の数が大きくてもその傾向は，顕著ではないが同様である．

ひとつのミーティングで多くの議題について情報交換の対象とすれば，交換される情報量も増える．たまたま PL 同士が出会っても，直近に PL から出たテーマのみを情報交換の対象とするのではなく，「そう言えば，あれは（過去のテーマ）どうなった？」と古いテーマに遡及して情報交換を行うことにより，さらにコミュニケーション効率が高くなることを意味している．

また、PL の数が大きいほど、テーマの遡及範囲を大きくする効果は高いことが分かる。これは、あとの節で詳述する。

#### 4.3.4 PM の指示サイクルとテーマの遡及

PM からの指示メールの送出サイクルとテーマの遡及範囲の交互作用についてシミュレーションを行った。実験条件は表 3 のとおりである。

表 3 指示メールの送出サイクルと知識交換テーマを遡る効果の交互作用実験パラメータの組み合わせ

パラメータ名	パラメータの値
PL の数 n	12
テーマの遡及範囲	1, 3, 5, 7, 9
指示メールの送出サイクル	1,2,3,4,5
留まるステップ数（ミーティング時間）	3

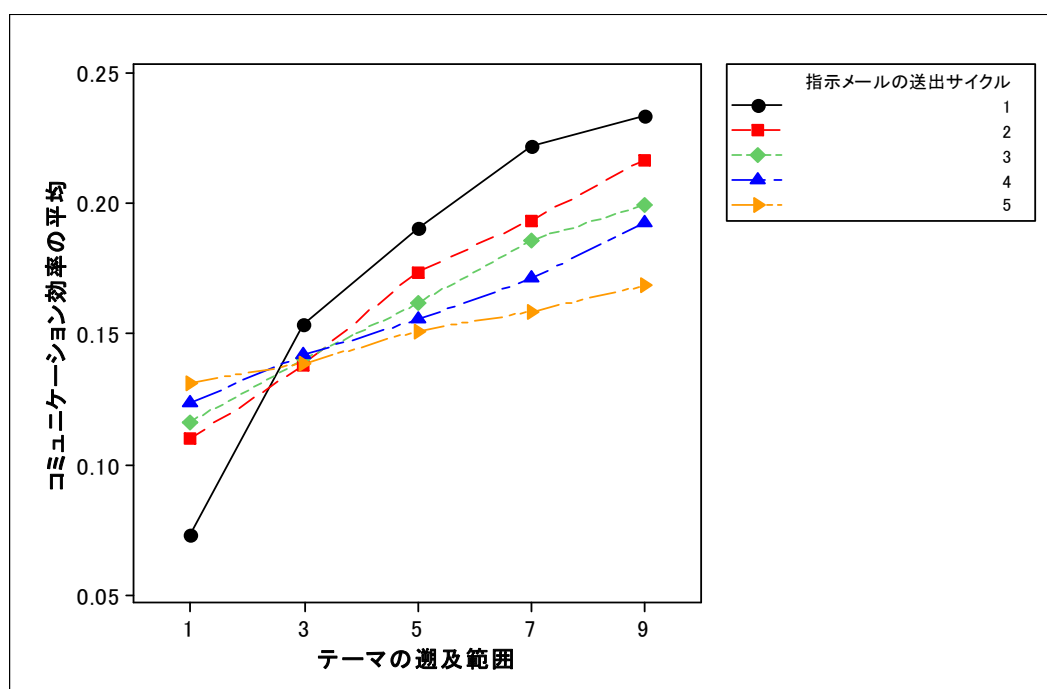


図 17 指示メールの送出サイクルから見たテーマの遡及範囲とコミュニケーション効率の関係

結果を図 17 に示す。横軸にテーマの遡及範囲（ミーティングの議題数にあ

たる), 縦軸にコミュニケーション効率を取った. シミュレーションは, 指示メールの送出サイクルを変えて 5 本の線にした. PL の数: 12. ミーティング時間: 3 ステップ. グラフの線はいずれも乱数シード値 10 回の平均値である.

#### 4.3.5 シミュレーション結果と考察: PM の指示サイクルとテーマの遡及

指示メールの送出サイクルに関わらず, テーマの遡及範囲 (対象となる議題の数) を大きくすると, コミュニケーション効率は上がる. コミュニケーションのテーマを古いテーマに遡って扱う効果は, 指示メールの送出サイクルが短いほど顕著である. テーマの遡及範囲が小さい範囲だと指示メールの送出サイクルが遅い方が, コミュニケーション効率は高い. しかし, 指示メールの送出サイクルが 1 の場合, その効果はテーマの遡及範囲が大きくなるにつれグラフが寝てくることから, 限度があることが分かる.

一回のミーティングにおいてテーマの遡及範囲 (対象となる議題の数) を多く取る効果は, 指示メールの送出サイクルが短いプロジェクトで著しい. つまり, PM がてきぱきと矢継ぎ早に PL に対して情報交換の指示を出しているプロジェクトほど, ミーティングで扱う議題の数を増やすと効果があがり, コミュニケーションの効率が良くなる.

但し, 指示メールの送出サイクルが 1 の場合, その効果はグラフが寝てくることから限度があることが分かる. 本シミュレーションのパラメータと現実プロジェクトのパラメータがそのまま同一ではない. 従って, どのあたりの指示メールの送出サイクルの値で限度が見えてくるのかは, 実際のプロジェクトにおける研究を待たなければならない.

もうひとつ図 17 から見える特徴は, 5 つの値の指示メールの送出サイクルの線がテーマの遡及範囲 3 あたりで交差していることである. 図 13 の結果は, テーマの遡及範囲を 1 に取って実験をしている. この実験をテーマの遡及範囲を 3 以上さらに大きく取って実験を行うことにより, 別の傾向を得る可能性がある.

#### 4.3.6 ミーティング時間とテーマの遡及

ミーティング時間とテーマの遡及範囲に関するシミュレーションを行った。実験条件は表4のとおりである。

表4 ミーティング時間と知識交換テーマを遡る効果の交互作用実験パラメータの組み合わせ

パラメータ名	パラメータの値
PL の数 n	12
テーマの遡及範囲	1, 3, 5, 7, 9
指示メールの送付サイクル	3
留まるステップ数 (ミーティング時間)	1,2,3,4,5

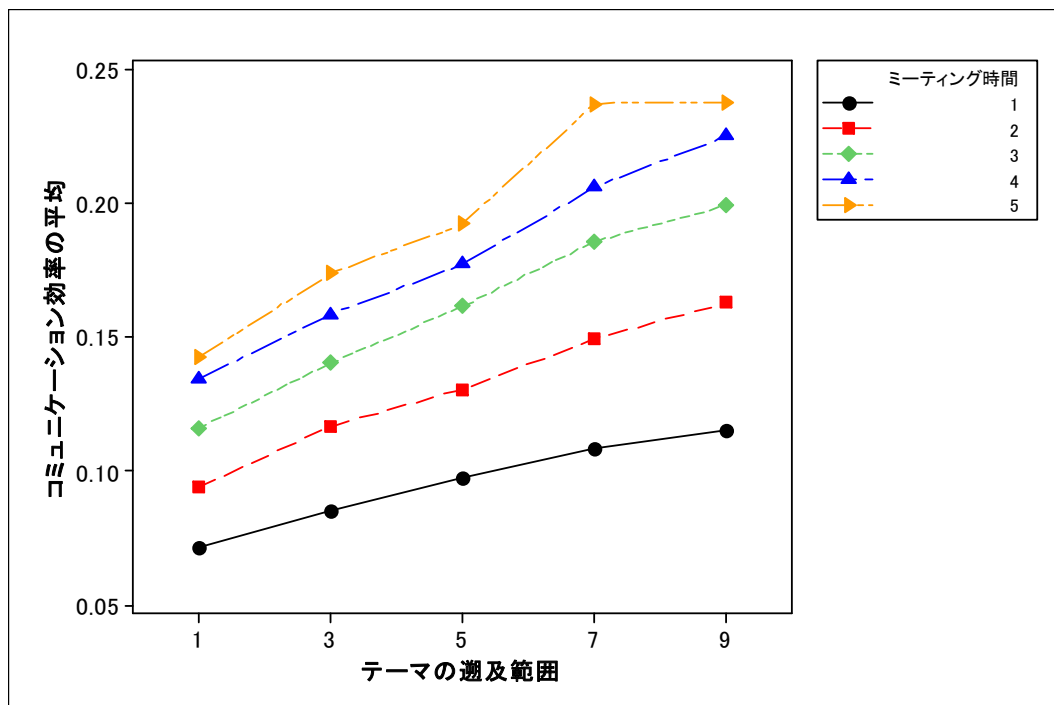


図 18 ミーティング時間から見たテーマの遡及範囲とコミュニケーション効率の関係

結果を図 18 に示す。横軸にテーマの遡及範囲 (ミーティングの議題数にあたる)、縦軸にコミュニケーション効率を取っている。シミュレーションは、ミーティング時間を変えて 5 本の線にしている。PL の数 : 12。指示メール

の送出サイクル：3 ステップ。グラフの線はいずれも乱数シード値 10 回の平均値である。

#### 4.3.7 シミュレーション結果と考察：ミーティング時間とテーマの遡及

テーマの遡及範囲を大きくするとコミュニケーション効率は高くなる。これは図 17 から既知である。このグラフから新たに分かることは、ミーティング時間を長くすることがコミュニケーション効率を上げることに寄与することである。ミーティング時間を長くするとは、シミュレーションで、PL エージェント同士が出会った際に、同じセルに留まるステップ数を長くすることを意味する。つまり、すぐに別れて新たな他の PL と出会い、新たな知識をもらう機会を犠牲にしても、一度出会った PL との知識交換を長くすることが良いことが分かる。もうひとつグラフから伺えるのは、ミーティング時間を増やしていった場合、徐々に線の間隔が狭くなっていることが分かる。ミーティングの時間を増やす効果は限度があることが分かる。

短いミーティングを多数持つよりも、ミーティング時間を長くし（PL が接触して留まるステップ数を増やす）、対象となる議題の数（テーマの遡及範囲）を多くすると、コミュニケーション効率が上がる。ミーティング時間を長くした方が、グラフの傾きが大きく、議題を多くする効果との相乗効果が認められる。例えば、現実の大規模プロジェクトで行われている合宿などは効果的であると思われる。ミーティングを細切れに短いミーティングを多く開催するのは効率的ではない。

実際のプロジェクトでも、なかなか時間調整がつかない PL 同士が、いったんミーティングを持った際に、出来るだけ時間を十分に取り、多くの議題をこなすことが効果的であることは想像がつく。

このような結果になった理由を考察する。ミーティング時間を長くすることは PL の接触確率を大きくすることになる。また、議題の数を多くすることは PM から指示のあったテーマの知識が PL 同士で異なる確率を高くすることになり、情報の交換が促進される。よって、ミーティング時間を長くし、議題を多くすることは、コミュニケーション効率を上げることに効果がある。但し、上述のように、ミーティング時間を長くする効果には限度があることが分かる。実際のプロジェクトに於いて、どれくらいが適切かについて、今

後事例研究で確認することは意義がある。

## 4.4 まとめ

PL の数  $n$  が増えると、コミュニケーション効率は減少する。つまり、プロジェクトの規模が大きくなるほどコミュニケーション効率が悪くなる。コミュニケーション効率が減少するという事は、シミュレーション終了時点での、プロジェクトチーム内の知識の拡散総量が小さいことを意味する。この結果は、コミュニケーション効率の式 (1) の分母に  $n^2$  が入っているため、予想された通りの結果となった。

コミュニケーション効率を上げる方法に関して、次のような結果が得られた、1) PM からの指示メールの間隔を長くするほど、コミュニケーション効率は高くなる。PM は、決められたプロジェクト納期の中で、適切な程度にゆっくりと指示を出すと良い。2) ミーティングの議題数を多く取ると、PL の数、指示メールの送出サイクルに関わらずコミュニケーション効率が上がる。議題数を大きくする効果は、指示メールの送出サイクルが短いほど顕著である。3) 短いミーティングを多数持つよりも、ミーティング時間を長くし、対象となる議題の数を多くすると、コミュニケーション効率が上がる。ミーティング時間を長くする効果と、議題を多くする効果との相乗効果が認められる。

また、今後、多大な実験量になると思われるが、PL の数  $n$ 、テーマの遡及範囲、留まるステップ数（ミーティング時間）の 3 値を同時に大きなレンジで変えての交互作用を確認することにより新たな知見が得られる可能性がある。

## 第5章 アンケート調査

### 5.1 はじめに

シミュレーション結果を現実プロジェクト遂行業務に携わる人たちの意識と比較するために、アンケート調査を行った。アンケート調査はプロジェクト遂行に従事している者に対して行われた。アンケートの質問内容は、プロジェクトにおけるコミュニケーションの負荷の大きさとプロジェクト規模とミーティングの持ち方についてである。

### 5.2 調査の方法

アンケート調査は、日本ヒューレット・パカード株式会社のシステムインテグレーション部門にて実施した。アンケートはプロジェクト遂行に従事している者が参加する Project Management Learning Community (PMLC)の参加希望者に対して行われた。

アンケートの回答は、PMLC 参加申し込みを社内イントラネットの WEB ページで行う際に、強制的に入力をさせる方法で実施された。PMLC 参加者の中にはプロジェクトに従事していないものも含まれるため、回答者の中でアンケート目的に合致する者の回答のみを結果として集計した。アンケートの質問項目、実施日、回答者数は、末尾付録に記載した。

### 5.3 プロジェクトにおけるコミュニケーションの負荷の大きさ

プロジェクト関係者に対して、アンケートにより自らの業務時間に占めるコミュニケーションの時間割合を主観的に回答してもらった。その結果を図 19 に示す。図 19 では、横軸にプロジェクトにおける職位を取っている。縦軸はプロジェクトでの仕事時間に占めるコミュニケーション時間の時間割合%を取った。

図 19 に示す内容から、統括 PM 及び PM はその業務時間のおおよそ半分をコミュニケーションに充てていることが分かる。PL も平均値でおおよそ 40%をコミュニケーションに充てており、業務時間における割合が高い。メンバ（プロジェクトにおける一般メンバ）は平均して 30%程度であり、上位

の職位に比べてコミュニケーションが占める時間割合は少し小さい。このように、プロジェクトチームにおける PL 以上の職位では、コミュニケーションの負荷が高いことが分かる。この結果は、プロジェクトにおける仕事の多くがコミュニケーションに使われているという多くの関連研究の内容と合致する。

各専門領域の業務を、プロジェクトの一つの目的に向かって整合を取っていくことがプロジェクト運営では求められている。統括 PM, PM, PL のコミュニケーション時間の多くは、そのことに費やされる。また、図 19 から分かるように、プロジェクトチームの上位職位においては、コミュニケーションの負荷が高い。このことから、コミュニケーションの効率の良否はプロジェクトの生産性に大きく影響していると推察される。本研究がコミュニケーション効率をテーマにする意義が確認された。

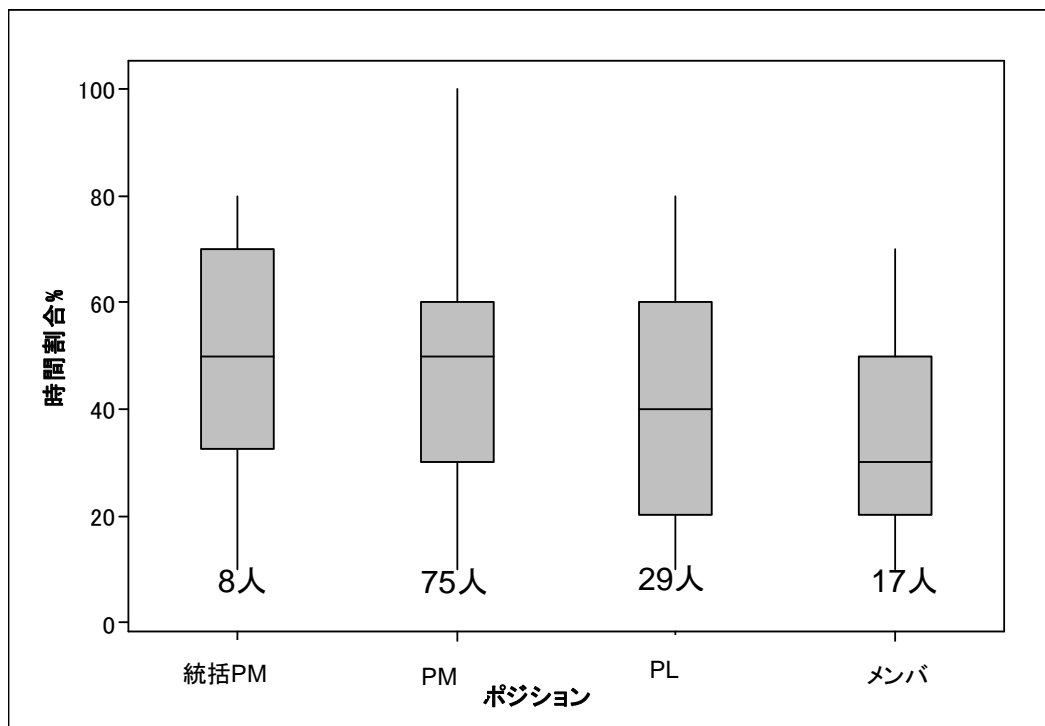


図 19 プロジェクトにおけるポジション別 業務時間に占めるコミュニケーション割合



## 5.4 プロジェクト規模とミーティングの持ち方

「効率の良いコミュニケーション方法」の結果と照合するために、プロジェクト規模とミーティングの持ち方についてアンケートを取った。ミーティング時間の長さ及び議題の多さについて聞いた。結果を図 20 に示す。

アンケートによる主観的対比の上での回答だが、プロジェクトの規模が大きくなるに従って、「どちらかと言うと、議題を多くして、長時間のミーティングを回数少なく開催した方が効率は良い」という意見の比率が高い。同時に、プロジェクト規模が大規模だと、「どちらかと言うと、議題を少なくして、短時間のミーティングを頻繁に開催した方が効率は良い。」は大幅に減っている。この結果は、本論文のシミュレーション結果と同じ傾向を示唆している。

一方、少なからず「下記のどちらでも効率はおなじである。」としている人がおり、この様な知見が完全に共有化されていないことが分かる。

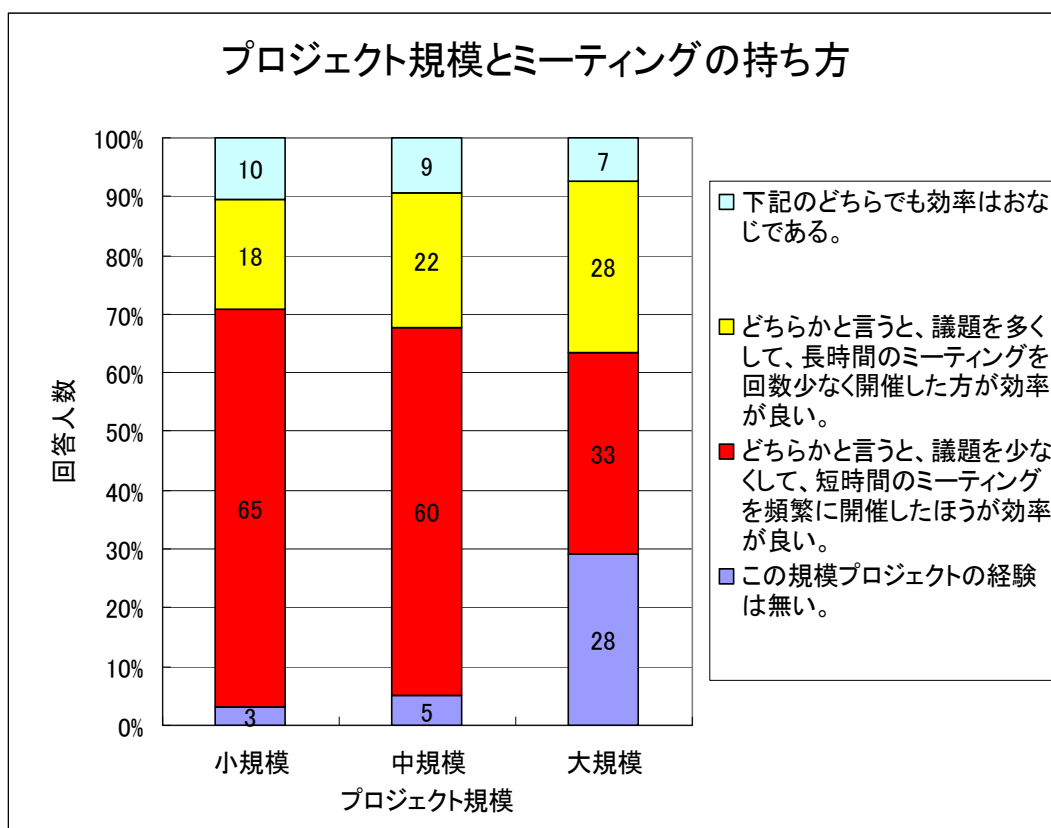


図 20 プロジェクト規模とミーティングの持ち方に対する意見

## 5.5 まとめ

PM, PLとも業務時間における割合が高い。一般メンバは上位の職位に比べてコミュニケーションが占める時間割合は少し小さい。このように、プロジェクトチームにおけるPL以上の職位では、コミュニケーションの負荷が高いことが分かる。このことから、コミュニケーションの効率の良否はプロジェクトの生産性に大きく影響していると推察される

プロジェクトの規模が大きくなるに従って、「どちらかと言うと、議題を多くして、長時間のミーティングを回数少なく開催した方が効率は良い」という意見の比率が高い。

## 第6章 結論

### 6.1 結論

第1章でプロジェクトの成功・失敗が、コミュニケーション効率によって左右されることを示し、第2章においてコミュニケーション効率の影響が大きい複雑な連携を伴うプロジェクトのチーム内コミュニケーションを本研究の対象と定義した。また、プロジェクトマネジメントに関する関連研究のサーベイにより、プロジェクトにおけるコミュニケーションの特性を整理し、エージェント・ベース・シミュレーションのモデルを単純化する根拠を得た。関連研究ではコミュニケーションについての測定評価尺度が定められていなかったが、本論文で新たに導入した「コミュニケーション効率」という評価尺度について定義・説明した。第3章で、現実のプロジェクトにおけるコミュニケーションのプロセスと関連研究から、シミュレーションモデルを設定した。第4章では、シミュレーションから得られた結果から、PLの数の影響、指示メールの送出サイクルの影響、効率の良いコミュニケーション方法（知識交換テーマを遡る効果、PMの指示サイクルとテーマの遡及の相互作用、ミーティング時間とテーマの遡及の相互作用）について考察を行った。また、第5章では、シミュレーション結果と現実とを対比するために行ったアンケート調査について報告した。

プロジェクト規模とコミュニケーションのあり方がコミュニケーション効率に与える影響と、コミュニケーション効率を上げるための知見について、本研究で得られた結論は下記のとおりである。

- 大規模プロジェクトは、小規模プロジェクトと比べると、コミュニケーション効率が低い。
- コミュニケーション効率を高くしたいならば、PMはプロジェクト納期の許す限り、ゆっくりとPLに指示を出した方が良い。
- PMが指示をてきぱき出す場合は、一回のミーティングで多くのテーマを議題にすると、コミュニケーション効率を効果的にあげることが可能である。
- ミーティングの回数を減らす代わりに、ミーティング時間を長くし、多くのテーマを議題にすると、コミュニケーション効率を高めることについて、相乗効果が認められる。（アンケート調査から、現場のプロジェクト関係者の一部に気が付いている人がいることが分かる。）

シミュレーションによって裏付けられた知見は、ソフトウェア開発プロジェクトの効率化とトラブル防止に役立つことにより、ソフトウェア開発企業の経営改善に資すると考える。

## 6.2 今後の課題

今後の課題として、次の二つが挙げられる。

### 1) コミュニケーション効率に影響を及ぼす因子のさらなる研究

本研究では、既往研究に無かった評価尺度：「コミュニケーション効率」を導入した。コミュニケーション効率については、更なる影響因子に関するシミュレーションにより、新たな知見が得られる可能性がある。今後の課題として、エージェント・ベース・シミュレーションのモデルに新たな条件を設定して、知見を広げることが望まれる。特に、PL の数、指示メールの送出周期、テーマの遡及範囲、留まるステップ数（ミーティング時間）の4値を同時に大きなレンジで変えての交互作用を確認したい。

### 2) この研究で得られた知見に関わる事例研究

プロジェクトマネジメントにおけるコミュニケーションについて、プロジェクトの知識体系である PMBOK[7]は、コミュニケーションに関わるプロセス以外あまり語っていない。また、プロジェクトマネジメントに関わる書籍にも、コミュニケーションの効率化に関する知見は少ない。

プロジェクトの現場では、コミュニケーションに関する経験則（知見）はプロジェクト関係者が **On the Job Training** で習得しているのが現状であり、公知化されていない。そのため、プロジェクト関係者は、多くの共有していない経験則がある。そのために、発生しなくてよいトラブルが発生しているケースが見られる。

本研究はプロジェクト関係者の中で何となく意識されているコミュニケーション効率に関わる知見の存在を示すことができた。この知見が、事例研究を行うことによって明確な裏付けがなされ、公知化されることが望まれる。

以上

## 謝辞

本論文は、東京工業大学総合理工学研究科知能システム科学専攻寺野隆雄教授のご指導のもとにとりまとめたものです。丁寧なご指導は、論文作成のためだけでなく、ビジネスの現場においてロジカルな考え方を習慣とすることに大変役立ちました。

論文の審査過程を通じて、出口 弘 先生、新田 克己 先生、小野 功 先生、吉川 厚 先生には貴重なコメントをいただき、論文の主張をはっきりさせることができました。また、東京工業大学寺野ゼミの皆様には、ゼミの議論の中で示唆に満ちたご意見をいただきました。特に、吉川先生には、論文の最終稿をまとめるにあたって貴重なアドバイスを頂き、筋道の通った論旨にすることができました。

論文を投稿させていただいた各学会・学会誌・本論文の査読を担当された匿名の先生方からは、多角的視点からの査読意見をいただき、研究における不適切な部分を修正することが出来ました。研究成果を引用させていただいた先行研究は、本論文の骨格を作る際に大変役に立ちました。感謝いたします。

また、エージェント・ベース・シミュレータ「*artisoc*」[10][24]を提供くださった株式会社構造計画研究所に感謝いたします。

最後に、励ましや助言をくれた家族、友人に心より感謝を込めてお礼申し上げます。

## 参考文献

- [1] 岡村正司「超大型プロジェクトでのプロジェクト管理技法の実践事例（タイム管理，コスト管理，変更管理，品質管理，コミュニケーション管理）」（プロジェクトマネジメント学会 2000 年度秋季研究発表大会予稿集 2000 年）
- [2] 伊東俊彦「ネットワーク組織のプロジェクトにおける電子コミュニケーションのマネジメント」（プロジェクトマネジメント学会誌 2005 年 Vol.7, No.1 pp.32-37）
- [3] 東基衛，細谷僚一，高橋宗雄「ソフトウェアマネジメント概説」（情報処理 1992 年 8 月 Vol.33, No.8 pp.894-905）
- [4] 後藤兼一，望月正嘉，寺山隆司「ソフトウェア開発において経済性を考慮したチーム人数の求め方」（情報処理学会論文誌 1984 年 3 月 Vol.25, No.2 pp.225-234）
- [5] 建部清美，阿部未寿希，関哲朗「CCPM によるソフトウェア開発プロジェクトの短納期化実現に関する研究」（プロジェクトマネジメント学会誌 2000 年 10 月 Vol.8, No.5）
- [6] 仲谷美江「ソフトウェア開発プロジェクトにおけるコミュニケーション支援に関する研究」（大阪大学博士論文 1995 年 8 月）
- [7] PMI「プロジェクトマネジメント知識体系ガイド第 3 版（PMBOK ガイド）」（Project Management Institute, Inc. 2004 年）
- [8] フレデリック・P・ブルックス，JR.「人月の神話 新装版 狼人間を撃つ銀の弾はない（新装版）」（ピアソン・エデュケーション 2006 年 11 月）
- [9] ロバート・アクセルロッド著，寺野隆雄監訳「対立と協調の科学 エージェント・ベース・モデルによる複雑系の解明」ダイヤモンド社 2003 年）
- [10] 山影 進「人工社会構築指南—artisoc によるマルチエージェント・シミュレーション入門」（書籍工房早山 2007 年）
- [11] Joshua M. Epstein, Robert Axtell 著，服部正太，木村香代子訳「人工社会 - 複雑系とマルチエージェント・シミュレーション-」（共立出版 1999 年）
- [12] Robert Axelrod, "Advancing the Art of Simulation in the Social Science", (Forthcoming in a special issue on agent-based modeling in the Japanese Journal for Management Information Systems (updated version of an article originally published in Rosario Conte, Rainer Hegselmann and Pietro Terana (eds.), Simulating Social Phenomena), (Springer-Verlag), Aug. 2003 (1997), pp.1-19 (pp.21-40))

- [13] Rebecca Zimoch, “Cooperation, communication and teamwork are key to project's success”, (Water Engineering & Management, Des Plaines, Jan. 2000 Vol.14, Iss.1 pp.14-17)
- [14] Ware Myers, “Early Communication Key to Software Project Success”, (Computer, May, 1999 pp.110-111)
- [15] Richard S. Woodward, “Preparing Your Project Team for Successful Outsourcing”, (BioPharm, May.1999 pp.58-60)
- [16] Liz Michalski, “Effective Communication Equals Successful Project Management”, (BioPharm, May.2000 pp.14-18)
- [17] B.T.Turner, “Project Engineering and the need for good communication”,(CME-Chartered Mechanical Engineer, June, 1983 Vol.30,6 pp.36-38)
- [18] Sharlett Gillard, Jane Johansen, “Project management communication: a systems approach”, (Journal of Information Science, 2004. 30 (1),pp. 23-29)
- [19] Mary Beth Pinto, Jeffrey K. Pinto, “Project Team Communication and Cross-Functional Cooperation in New Program Development”, (J. Prod. Innov. Manag. , 1990, pp.200-212)
- [20] Mie Nakatani, Shogo Nishida, “Trouble Communication Model in a Software Development Project”, (IEICE TRANS. FUNDAMENTALS, Feb. 1992, Vol.E75-A, No. 2 pp.196-206)
- [21] ISBSG, “The impact of team size on productivity and output”, (Special Analysis Report, International Software Benchmarking Standards Group (ISBSG), 2007)
- [22] ISBSG, “Practical Project Estimation 2nd Edition”, (International Software Benchmarking Standards Group (ISBSG), 2005)
- [23] PMI東京支部ホームページ, <<http://www.pmi-tokyo.org/02/000200.html>>
- [24] MAS コミュニティ (構造計画研究所) サンプルモデル <<http://mas.kke.co.jp/modules/mydownloads/>>
- [25] 相良博喜ら「マルチエージェント・シミュレーションを適用したプロジェクトマネジメントのモデル化」(日本計算工学会論文集 2005年 Volume7,2005 pp.159-164)
- [26] 細谷和伸ら「プロジェクトを成功に導くためのノウハウの展開」(プロジェクトマネジメント学会誌 2005年 Vol.7,No.6 pp. 13-15)
- [27] コミュニケーション・マネジメント研究会「プロジェクトマネジメントにおけるコミュニケーション設計」(プロジェクトマネジメント学会誌 2005年 Vol.7,No.3 pp.44-49)
- [28] 藤田芳夫「プロジェクトマネジメントとは」(情報の科学と技術 2003年

- 53 卷 5 号 pp.236-241)
- [29] 青木一三「プロジェクトのカオス現象」(プロジェクトマネジメント学会 2000 年度秋季研究発表大会予稿集 2000 年 pp.205-210)
  - [30] 長瀬勝彦「プロジェクトチームの生産性」(駒大経営研究 1998 年 Vol.29, No.3/4 pp.105-117)
  - [31] 近藤洋司「プロジェクト成功法則と成功確率向上への提言」(プロジェクトマネジメント学会誌 2005 年 Vol.7, No.6 pp.9-12)
  - [32] 森川勇治「プロジェクトが失敗する本当の理由」(Computer Report 2007 年 2007/5 pp.20-24)
  - [33] 町田仁司「プロジェクト・リスク・マネジメント 失敗プロジェクトには共通項が多い-要求仕様の鵜呑みは危険」(日経システム構築 2003 年 No.127 日経 BP 社 pp.148-159)
  - [34] 遠藤雄一「プロジェクトマネージャのためのコミュニケーション・マネジメント (体系と Lessons Learned)」(ProVISION 2003 年 No.38 pp.44-53)
  - [35] 小松昭英「プロジェクト・ブレイクダウン・ストラクチャ」(プロジェクトマネジメント学会誌 2004 年 Vol.6, No.5 pp.45-49)
  - [36] 江熊きよみ「プロジェクトチームの生産性に影響を及ぼす要因」(プロジェクトマネジメント学会 2002 年度春季研究発表大会予稿集 2002 年 pp.46-48)
  - [37] 治田倫男「プロジェクト・チーム・ライフ・サイクルにおける効果的なリーダーシップ・スタイルの事例研究」(プロジェクトマネジメント学会誌 2006 年 Vol.8, No.6 pp.23-28)
  - [38] 関根清隆「悩み多きプロジェクトマネージャへ贈る言葉」(プロジェクトマネジメント学会誌 2000 年 Vol.2, No.2 pp.13-18)
  - [39] 実森仁志「特集 3 成功するプロジェクト推進術-三つの“コミュニケーション技術”を活かそう」(日経システム構築 2006 年 No.154 日経 BP 社 pp.42-51)
  - [40] 秋山雅俊「特集「コミュニケーション・マネジメント」に寄せて」(プロジェクトマネジメント学会誌 2002 年 Vol.4, No.32)
  - [41] 松島桂樹「特集「コミュニケーション・マネジメント」に寄せて」(プロジェクトマネジメント学会誌 2005 年 Vol.7, No.12)
  - [42] 横島文雄ら「チームビルディングとコミュニケーション・ツール」(プロジェクトマネジメント学会 2001 年度秋季研究発表大会予稿集 2001 年 pp.168-170)
  - [43] 治田倫男「チーム運営におけるコミュニケーションの重要性」(プロジェクトマネジメント学会誌 2005 年 Vol.7, No.1 pp.3-8)



- [44] 長谷川淳「大規模・短納期システム開発でのPMのポイント」(プロジェクトマネジメント学会 2003 年度春季研究発表大会予稿集 2003 年 pp.256-258)
- [45] 込山俊博ら「ソフトウェアプロセスにおけるコミュニケーションの側面とパートナー満足」(プロジェクトマネジメント学会誌 2002 年 Vol.4,No.3 pp.24-27)
- [46] 高木徳生「ソフトウェア中心の大規模システム開発プロジェクト事例にみる成功要因と考察」(情報処理 2003 年 4 月 44 巻 4 号 pp.348-356)
- [47] 服部昇ら「ソフトウェア開発プロジェクトの大きさと生産性の関係に関する考察」(プロジェクトマネジメント学会誌 2006 年 Vol.8,No.4 pp.13-16)
- [48] 長谷川浩道ら「ソフトウェア開発プロジェクトにおける最適な情報マネジメントの検討」(情報処理学会研究報告 2007 年 2007-GN-64(2) pp.7-12)
- [49] 中谷美江ら「ソフトウェア開発プロジェクトにおけるインフォーマルコミュニケーションの評価手法」(電子情報通信学会論文誌 1994 年 Vol.J77-D-II No.4 pp.823-837)
- [50] 会田信弘「組織マネジメントとコミュニケーション」(UNISYS Technology Review 2000 年第 67 号 pp.184-207)
- [51] 岡田清久「事例分析に基づく失敗プロジェクトの研究」(プロジェクトマネジメント学会誌 2006 年 Vol.8,No.6 pp.42-46)
- [52] 内田吉宣「失敗知識を用いたリスクマネジメント支援方法の提案」(プロジェクトマネジメント学会誌 2005 年 Vol.7,No.6 pp.3-8)
- [53] 村松充雄「システム開発プロジェクトに関わるリスクマネジメント」(プロジェクトマネジメント学会誌 2007 年 Vol.9,No.4 pp.18-22)
- [54] 岡田公治「事業部門横断的なプロジェクトマネジメント業務ナレッジ抽出の試み」(プロジェクトマネジメント学会誌 2008 年 Vol.10,No.3 pp.23-28)
- [55] 松澤芳昭ら「産学協同によるプロジェクトマネージャ育成システムの提案と実証実験」(情報処理学会論文誌 Mar.2007 Vol.48,No.3 pp.976-987)
- [56] 久井信也「コミュニケーションの原点を探る」(プロジェクトマネジメント学会誌 2004 年 Vol.6,No.3 pp.43-45)
- [57] 山本吉伸ら「コミュニケーション基礎実験計画とその環境設計」(ヒューマンインターフェース 1995 年 60-7 pp.49-56)
- [58] 嶋野淳「個別+連結会計プロジェクトをわずか 1 年で完遂」(ソリューション IT 2004 年 2004.6 pp.26-34)
- [59] コミュニケーション・マネジメント研究会「企業情報システムプロジェクトのコミュニケーション・ブレイクダウン・ストラクチャ」(プロジェクトマネジメント学会誌 2006 年 Vol.8,N0.5 pp.4-9)

- [60] 山崎義雄ら「企業グループ標準会計システムの構築」(プロジェクトマネジメント学会 2003 年度春季研究発表大会予稿集 2003 年)
- [61] 石倉政幸「海外大型生産設備プロジェクトの運営 -カタールガス LNG プラント建設プロジェクト-」(プロジェクトマネジメント学会 2000 年度秋季研究発表大会予稿集 pp.191-196)
- [62] 大坂宏「海外大型建設プロジェクトのコミュニケーション管理」(プロジェクトマネジメント学会誌 2004 年 Vol.6,No.4 pp.58-61)
- [63] 大黒英和「SI プロジェクトにおけるリスクマネジメント」(プロジェクトマネジメント学会誌 2004 年 Vol.6,No.4 pp.28-32)
- [64] 松島桂樹「PM におけるコミュニケーション・マネジメント研究の視座」(プロジェクトマネジメント学会誌 2004 年 Vol.6,No.2 pp.46-50)
- [65] コミュニケーション・マネジメント研究会「PM における効果的コミュニケーション・マネジメントのフレームワーク第 1 回」(プロジェクトマネジメント学会誌 2005 年 Vol.7, No.1 pp.38-41)
- [66] 竹久友二「PM コミュニケーション考」(プロジェクトマネジメント学会誌 2001 年 Vol.3,No.6 p.1)
- [67] 出口弘「P2M 思考にとってのエージェント&ゲーミングシミュレーション」(研究技術計画 2005 年 21 (2) pp.170-175)
- [68] 倉谷祥久「3 トップフォーメーション」(プロジェクトマネジメント学会 2001 年度秋季研究発表大会予稿集 2001 年 pp.7-12)
- [69] Robert W. Root “Design of a Multi-Media Vehicle for Social Browsing” (Proceedings of CSCW '88 1988 pp.25-38)
- [70] S.R.Thomas, R.L.Tucker and W.R.Kelly “Compass: An Assessment Tool for Improving Project Team Communications” (Project Management J. Dec.1999 Vol.30, No.4 pp.15-24)
- [71] Marly M. Carvalho “Communication Issues in Projects Management” (PICMET 2008 Proceedings 27-31 July, 2008 Cape Town pp.1280-1284)
- [72] Robert Axtell “Why Agents? On the varied motivations for agent computing in the social sciences” (Center on Social and Economic Dynamics Nov., 2000 Working Paper No. 17 pp.1-22)
- [73] Jan Terje Karlsen, Jeanette Andersen, Live S. Birkely “An empirical study of critical success factors in IT projects” (Int. J. Management and Enterprise Development 2006 Vol.3, No.4 Inderscience Enterprises Ltd. pp. 297-311)
- [74] 日経コンピュータ「特集 第 2 回 プロジェクト実態調査 800 社 成功率は 31.1%」2008 年 No.718 pp.36-53

# 研究業績一覧

## 学術論文

- 「プロジェクトマネージャの指示遅れによる生産性へのインパクト」, 2008年9月, 経営情報学会誌, Vol.17 No.2 Sep.2008 pp.21-pp.38, (渡部雅男/寺野隆雄), 査読あり
- “Agent-Based Simulation to Seek For Effective Communication In Project Management”. 2007年9月, Journal of Convergence Information Technology Vol.2, Number 3, September 2007 pp.39-46, (渡部雅男/寺野隆雄), 査読あり
- 「自律分散オペレーションの効果的応用」, 2003年12月, プロジェクトマネジメント学会誌 Vol.5 No.6 pp.36-pp.42, (渡部雅男), 査読あり

## 学会発表・プロシーディング

- “Practical Communication in Large Scale Projects”, 2007年8月, The 51st Annual Meeting of the International Society for the Systems Sciences p.119, (渡部雅男/寺野隆雄), 査読なし
- 「大規模プロジェクトにおけるコミュニケーション効率の特性と短納期化」, 2007年3月, プロジェクトマネジメント学会 2007年度春季研究発表会大会予稿集 pp.198-pp.203, (渡部雅男/寺野隆雄), 査読なし
- 「プロジェクトマネジメントにおける自律分散オペレーションの可能性」, 2003年3月, プロジェクトマネジメント学会 2003年度春季研究発表会予稿集 pp.102-pp.107, (渡部雅男), 査読なし
- “Project Management know-how in the procedures and manuals”, 2001年11月, International Project Management Congress 2001 p.16, (渡部雅男), 査読なし

## 雑誌掲載

- 「プラントにおける西暦 2000 年問題」, 1999年9月, 月刊高圧ガス平成 11年 VOL.36 pp.50-pp.51, (渡部雅男), 査読なし
- 「ハウステンボス・プロジェクト-ホライズン・アドベンチャーの水演出について」, 1993年9月, 千代田技報 1993年9月号 pp.72-pp.75, (渡部雅男/彦坂英敏/宍戸俊介/新井邦夫), 千代田化工建設内の査読あり

# 付録

## 付録 1：コミュニケーション負荷に関わるアンケート

2008年2月25日開催 PM Learning Community 参加 WEB 申込アンケート

対象者：日本ヒューレット・パッカー社員

回答者：155人。

質問内容：

1. 社員番号
2. 姓・名
3. 所属部署（選択）
4. 参加会場を選んで下さい（選択）：「東京会場」、「大阪会場」。
5. 現在、プロジェクトに参加しているか、過去3年以内にプロジェクトに参加していましたか？（選択）：「はい、プロジェクトに参加しています。もしくはしていました。」、「いいえ、プロジェクトに参加していません。」
6. 過去3年以内でもっとも最近参加したプロジェクトの受注金額規模を、下記の中から、ラジオボタンで選択してください。（選択）：「10億円以上」、「5億円以上、10億円未満」、「5千万円以上、1億円未満」、「1千万円以上、5千万円未満」、「1千万円未満」。
7. 過去3年以内でもっとも最近参加したプロジェクトでの役割を、ラジオボタンでお選び下さい。（選択）：「統括PM」、「PM」、「PL」、「プロジェクト内PMOもしくは事務局メンバ」、「それ以外のメンバ」。
8. 過去3年以内でもっとも最近参加したプロジェクトにおいて、プロジェクトを進めるために行ったコミュニケーションに掛かった時間の割合は、ご自分の全作業時間内のおおよそ何%を締めましたか、もしくは締めていますか？ ラジオボタンでお選び下さい。（選択）：「10%」「20%」「30%」「40%」「50%」「60%」「70%」「80%」「90%」「100%」。
9. 一般的に言って、大規模プロジェクトと小規模プロジェクトを比べて、どちらが、より短納期化しやすいと思いますか？ ラジオボタンでお選び下さい。（選択）：「小規模プロジェクトの方が短納期化しやすいと思います。」、「大規模プロジェクトの方が短納期化しやすいと思います。」、「プロジェクトの規模に関係は無いと思います。」。
10. 大規模プロジェクトと小規模プロジェクトを比べた場合、どちらの方が、必要な情報がプロジェクトチーム全体に伝わりにくいと思いますか？ ラジオボタンでお選び下さい。（選択）：「大規模プロジェクトの方が伝わりにくい。」、「小規模プロジェクトの方が伝わりにくい。」

## 付録 2：効率的なコミュニケーションに関わるアンケート

2008年6月13日開催 PM Learning Community 参加 WEB 申込アンケート

対象者：日本ヒューレット・パッカー社員

回答者数：160人

質問内容：

1. 社員番号
2. 姓・名
3. 属部署を選択してください。(選択)
4. 参加会場を選んで下さい。(選択)：「東京会場」, 「大阪会場」.
5. 現在のあなたのキャリアパスをお選び下さい。(選択)：「AS」, 「BPC」, 「PM」, 「SA」, 「TS」, 「その他」.
6. 現在のあなたの役割をお知らせ下さい。(選択)：「統括 PM」, 「PM」, 「PL」, 「プロジェクト一般メンバ」, 「EPMO」, 「プロジェクト内 PMO」, 「PMを配下に持つ箱長」.
7. 過去から, 現在までにプロジェクト遂行に直接関わった経験年数を教えてください。(年数入力)
8. 現在, 関わっているプロジェクトの金額規模をお答え下さい。(選択)：「10億円以上」, 「5億円以上, 10億円未満」, 「5千万円以上, 1億円未満」, 「1千万円以上, 5千万円未満」, 「1千万円未満」, 「該当なし」.
9. 現在, 関わっているプロジェクトの PL (PM の下のプロジェクトリーダー) の人数を教えてください。(選択)：「1人」, 「2人」, 「3人」, 「4人」, 「5人」, 「6人」, 「7人」, 「8人」, 「9人」, 「10人以上」, 「PL ないない」, 「現在, プロジェクトに参加していない」.
10. 「大規模」プロジェクトの場合の, プロジェクト内ミーティングの持ち方について, ご経験からお答え下さい。(選択)：「どちらかと言うと, 議題を多くして, 長時間のミーティングを回数少なく開催した方が効率は良い」, 「どちらかと言うと, 議題を少なくして, 短時間のミーティングを頻繁に開催したほうが効率は良い」, 「上記のどちらでも効率はおなじである」, 「大規模プロジェクトの経験は無い」.
11. 「中規模」プロジェクトの場合のプロジェクト内ミーティングの持ち方について, ご経験からお答え下さい。(選択)：「どちらかと言うと, 議題を多くして, 長時間のミーティングを回数少なく開催した方が効率は良い」, 「どちらかと言うと, 議題を少なくして, 短時間のミーティングを頻繁に開催したほうが効率は良い」, 「上記のどちらでも効率はおなじである」, 「中規模プロジェクトの経験は無い」.

12. 「小規模」プロジェクトの場合のプロジェクト内ミーティングの持ち方について、ご経験からお答え下さい。(選択) : 「どちらかと言うと、議題を多くして、長時間のミーティングを回数少なく開催した方が効率は良い.」, 「どちらかと言うと、議題を少なくして、短時間のミーティングを頻繁に開催したほうが効率は良い.」, 「上記のどちらでも効率はおなじである.」, 「小規模プロジェクトの経験は無い.」.

## 付録 3 : ABS のソースコード

```
Universe のルール :

Univ_Init{
//変数を定義する.
Dim myAgt0 As Agt
Dim myAgt1 As Agt
Dim myMgr0 As Agt
Dim myMgr1 As Agt
Dim myAgtSet As AgtSet
Dim myMgrSet As AgtSet

Dim i As Integer
Dim myStaffSet As AgtSet
Dim myStaff As Agt

//PL0 に対する設定 :
//Layer=0 を接触空間とし, PL を生成する.
For i=0 To Universe.PL の数 -1
myAgt0 = CreateAgt(Universe.PROJECT.PL0)
//全ての PL0 の接触知識を初期化する関数を起動する.
myAgt0.知識 = PL0 の接触知識の初期化(i)
Next i

//Layer=0 の接触空間に, PL をランダムに配置する.
MakeAgtSet(myAgtSet, Universe.PROJECT.PL0)
RandomPutAgtSet(myAgtSet)

//PM0 に対する設定 :
//Layer=0 の接触空間に, PM0 を定義する.
myMgr0 = CreateAgt(Universe.PROJECT.PM0)

//PM0 は接触による知識交換をしないので, この次元で
の知識を持たない.

//Layer=0 の接触空間に, PM0 をランダムに配置する.
MakeAgtSet(myMgrSet, Universe.PROJECT.PM0)
RandomPutAgtSet(myMgrSet)

//PL1 に対する設定 :
For Each myAgt0 In myAgtSet
//Layer0 の PL0 に対して, Layer1 に PL1 を対応させて生
成する.
myAgt1 = CreateAgt(Universe.PROJECT.PL1)
myAgt0.関係する PL = myAgt1
myAgt1.関係する PL = myAgt0
myAgt1.Layer = 1

//Layer=1 (e-mail 空間) の Layer=0 と同じ場所(X,Y)座標
に, PL を位置づける.
myAgt1.X = myAgt0.X
myAgt1.Y = myAgt0.Y

//全ての PL1 の e-mail による受信知識を初期化する関数
を起動する.
myAgt1.知識 = PL1 のメール知識の初期化(i)
Next myAgt0

//PM1 に対する設定 :
For Each myMgr0 In myMgrSet
//Layer0 の PM0 に対して, Layer1 に PM1 を対応させて生
成する.
myMgr1 = CreateAgt(Universe.PROJECT.PM1)
myMgr0.関係する PM = myMgr1
myMgr1.関係する PM = myMgr0
myMgr1.Layer = 1

//Layer=1 (e-mail 空間) の Layer=0 と同じ場所(X,Y)座標
に, PM を位置づける.
myMgr1.X = myMgr0.X
myMgr1.Y = myMgr0.Y

//全ての PM1 の e-mail で送る知識の初期化をする関数を
起動する.
myMgr1.知識 = PM1 のメール知識送付の初期化(i)
```

```

Next myMgr0

//関係する PM の指定, PL1 への線を引く.
MakeAgtSet(myStaffSet, Universe.PROJECT.PL1)
For Each myStaff In myStaffSet
AddAgt(myMgr1.PL1 の集合, myStaff)
myStaff.関係する PM = myMgr1
Next myStaff

//その他変数の初期化
//エージェント間の相互作用の数 (グラフに表示される)
を初期化する.
Universe.相互作用の数 = 0

//終了を判定するフラグを初期化する.
Universe.終了判定フラグ = True
}

Univ_Step_Begin{

}

Univ_Step_End{
//変数を定義する.
Dim myAgt0 As Agt
Dim myAgtSet As AgtSet
Dim 幅 As Integer
Dim 高さ As Integer
Dim 個別合計 As Integer
Dim 知識の総数 As Integer
Dim c As Integer
Dim 効率 As Double

//「PROJECT」空間の幅を数値化する.
幅 = GetWidthSpace(Universe.PROJECT)

//「PROJECT」空間の高さを数値化する.
高さ = GetHeightSpace(Universe.PROJECT)

//PL0 をエージェントセットにする.
MakeAgtSet(myAgtSet, Universe.PROJECT.PL0)

//終了するかどうかを終了判定フラグを見て判定する. 判定は, PL0 のルールの「Sub 終了条件チェック()」で行う.
If Universe.終了判定フラグ == False Then

//PL0 の知識の総数を計算する.
//知識の総数を初期化する.
知識の総数 = 0
For Each myAgt0 in myAgtSet
//各 PL0 の知識の合計を求める.
//まずは初期化から.
個別合計 = 0
For c = 1 to Universe.PL の数
個別合計 = 個別合計 + CInt(Mid(myAgt0.知識, c, 1))
Next c
//次に, 全ての PL0 の知識の合計を求める.
知識の総数 = 知識の総数 + 個別合計
Next myAgt0

//シミュレーションを終了し, コンソール画面に下記の情報を書き出す.
//効率を計算しておく.
効率 = (Round (1000 * (知識の総数 - 9*Universe.PL の数) / GetCountStep() / Universe.PL の数)) / 1000//行を替える.
PrintLn("")

ExitSimulationMsg("幅: "& 幅 & ", "& 高さ: "& 高さ & ", "& PL の数: "& Universe.PL の数 & " 記憶遡及範囲: "& Universe.記憶遡及範囲 & " メールの送出周期: "& Universe.メールの送出周期 & " 留まるステップ数: "& Universe.留まるステップ数 & " シミュレーション

```



```
ヨンステップ:" & GetCountStep() & " 知識の総数:" &  
知識の総数&" 効率:"&効率)
```

```
End If  
}
```

```
Univ_Finish{
```

```
}
```

```
//Universe_Init で指定したように、全ての PL0 の接触知識  
を初期化する。
```

```
function PL0 の接触知識の初期化(n As Integer) As String
```

```
{
```

```
//変数を定義する。
```

```
Dim str 接触知識 As String
```

```
Dim 接触初期知識 As String
```

```
Dim i As Integer
```

```
//PL0 が最初に持っている知識の数をコントロールパネ  
ルの「PL の数」と同じにし、全て 0 に初期化する。
```

```
str 接触知識 = ""
```

```
For i = 1 to Universe.PL の数
```

```
str 接触知識 = str 接触知識 & CStr(0)
```

```
Next i
```

```
//一人の PL0 について、各分野の専門家として、一分野の  
みレベル 9 の知識を持たせ、残りの分野のレベルを 0 と  
する。
```

```
接触初期知識 = Left(str 接触知識, n) & CStr(9) & Left(str  
接触知識, Universe.PL の数 - 1 - n)
```

```
//初期設定した知識を PL0 に付与する。
```

```
return(接触初期知識)
```

```
}
```

```
//Universe_Init で指定したように、全ての PL1 のメール知  
識を初期化する。
```

```
function PL1 のメール知識の初期化(n As Integer) As String
```

```
{
```

```
//変数を定義する。
```

```
Dim str メール知識 As String
```

```
Dim i As Integer
```

```
//PL1 が最初に持っている知識の数をコントロールパネ  
ルの「PL の数」と同じにし、全て 0 に初期化する。
```

```
str メール知識 = ""
```

```
For i = 1 to Universe.PL の数
```

```
str メール知識 = str メール知識 & CStr(0)
```

```
Next i
```

```
//初期設定した知識を PL0 に付与する。
```

```
return(str メール知識)
```

```
}
```

```
//Universe_Init で指定したように、全ての PM1 のメール  
知識を初期化する。
```

```
function PM1 のメール知識送付の初期化(n As Integer) As
```

```
String
```

```
{
```

```
//変数を定義する。
```

```
Dim strPM のメール知識 As String
```

```
Dim i As Integer
```

```
//PM1 の初期知識を初期化する。全ての知識をレベル 9  
に設定。
```

```
strPM のメール知識 = ""
```

```
For i = 1 to Universe.PL の数
```

```
strPM のメール知識 = strPM のメール知識 & CStr(9)
```

```
Next i
```

```
//初期設定した知識を PM1 に付与する。
```

```

return(strPM のメール知識)
}

PL0 のルール :
Agt_Init{
//隣人 PL と出会った時に、知識交換のために隣人と同じ
場所に留まるステップ数のカウンターを初期設定する。
My.Counter = Universe.留まるステップ数

//各 PL0 は、とりあえずランダム方向 1 へ一歩進む。
「PROJECT」空間の壁にぶつかったら、元来た方向に跳
ね返る。
Turn(rnd() * 360)
If forward(1) > 0 Then
Turn(180)
End If
}

Agt_Step{
Dim j as Integer
Dim k as Double
Dim myAgt As Agt
Dim 直進のステップ As Integer

直進のステップ = 5

//「直進のステップ」で定めるステップだけ直進を継続す
るかどうかを判定する数字を作る。
//一歩進み、ランダムに方法転換すると、同じ場所をぐる
ぐる回る傾向があるため、これを排除する。
j = GetCountStep()
k = j / 直進のステップ - Int(j / 直進のステップ)

//隣に PL0 がいれば、隣人となる。
MakeOneAgtSetAroundOwn(my. 隣 人 , 1,

```

```

Universe.PROJECT.PL0, False)

//隣人がいるかどうかの判定を行う。
If 1 <= CountAgtSet(my.隣人) Then
//隣人がいるので、全ての隣人と知識を交換し合う。
If My.Counter > 0 Then
//カウンターの数が 1 以上なので、相互作用を行い、知識
交換をする。
For Each myAgt In my.隣人
//隣人との知識交換の相互作用を別に定める「相互作用」
関数で行う。
my.知識 = 相互作用(my.知識, myAgt.知識)
//カウンターをひとつ引き、知識交換の時間(ステップ数)
がひとつ過ぎたことにする。
My.Counter = My.Counter - 1
//いろいろな色に PL を染める。(但し、出力設定で無色に
している場合を除く。)
My.色 = CInt(My.知識) * 150
Next my.Agt

Else
//カウンターの数が留まる期間が過ぎていることを示す
ので、カウンターを初期値に戻し、ランダム方向に散る。
My.Counter = Universe.留まるステップ数
Turn(rnd() * 360)

//すぐに同じ相手とくっつかないように、2 歩飛び跳ねる。
「組織」空間の壁にぶつかったら、元来た方向に跳ね返
る。
If forward(2) > 0 Then
Turn(180)
End If

End If

Else

```

```

//隣人がいないので、知識交換をせずに動き回る。
If k == 0 Then
//直進継続を判定する数字がゼロなので、ランダムな方向
に一步進む。
Turn(rnd() * 360)
//「組織」空間の壁にぶつかったら、元来た方向に跳ね返
る。
If forward(1) > 0 Then
Turn(180)
End If

Else
//直進継続を判定する数字がゼロでないので、そのまま一
歩直進する。
//「PROJECT」空間にぶつかったら、元来た方向に跳ね
返る。
If forward(1) > 0 Then
Turn(180)
End If
End If
End If

}

//隣人と異なる知識をひとつ隣人に近づけて向上させる。
function 相互作用(myStr As String, yourStr As String) As
String
{
Dim 新知識 As String
Dim 高い知識 As String
Dim i As Integer
Dim hitN As Integer
Dim 異知識の位置(100) As Integer
Dim pos As Integer
Dim 相互作用の位置 As Integer
Dim 一つ前の知識送出位置 As Integer

```

```

Dim 二つ前の知識送出位置 As Integer
Dim 三つ前の知識送出位置 As Integer
Dim 四つ前の知識送出位置 As Integer
Dim 五つ前の知識送出位置 As Integer
Dim 六つ前の知識送出位置 As Integer
Dim 七つ前の知識送出位置 As Integer
Dim 八つ前の知識送出位置 As Integer
Dim 九つ前の知識送出位置 As Integer
Dim 十前の知識送出位置 As Integer

//隣人と異なる知識の数の変数をリセットする。
hitN = 0

//相互作用させる（知識を与える）異知識の場所を「記憶
遡及範囲」に従って決める。
//PM1の「確定した送出種類の位置」から送出の「記憶」
を呼び出す。
//但し、最初は、X前の知識送出位置が0なので、その場
合は、99にする。
一つ前の知識送出位置 =
GetHistory(Universe.PROJECT.PM1.確定した送出種類の位
置, 1)
二つ前の知識送出位置 =
GetHistory(Universe.PROJECT.PM1.確定した送出種類の位
置, 2)
三つ前の知識送出位置 =
GetHistory(Universe.PROJECT.PM1.確定した送出種類の位
置, 3)
四つ前の知識送出位置 =
GetHistory(Universe.PROJECT.PM1.確定した送出種類の位
置, 4)
五つ前の知識送出位置 =
GetHistory(Universe.PROJECT.PM1.確定した送出種類の位
置, 5)
六つ前の知識送出位置 =
GetHistory(Universe.PROJECT.PM1.確定した送出種類の位
置, 6)

```

```

置, 6)
七つ前の知識送出位置 =
GetHistory(Universe.PROJECT.PM1.確定した送出種類の位置, 7)
八つ前の知識送出位置 =
GetHistory(Universe.PROJECT.PM1.確定した送出種類の位置, 8)
九つ前の知識送出位置 =
GetHistory(Universe.PROJECT.PM1.確定した送出種類の位置, 9)
十前の知識送出位置 =
GetHistory(Universe.PROJECT.PM1.確定した送出種類の位置, 10)

//「記憶遡及範囲」により、範囲分遡ってメール受領位置
//のみの知識を交換するよう相互作用する候補位置を決める。
If Universe.記憶遡及範囲 == 10 Then
If 十前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 十前の知識送出位置, 1),
Mid(yourStr, 十前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 十前の知識送出位置
hitN = hitN + 1
End If
End If

If 九つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 九つ前の知識送出位置, 1),
Mid(yourStr, 九つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 九つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 八つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 八つ前の知識送出位置, 1),
Mid(yourStr, 八つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 八つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 七つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 七つ前の知識送出位置, 1),
Mid(yourStr, 七つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 七つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 六つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 六つ前の知識送出位置, 1),
Mid(yourStr, 六つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 六つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 五つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 五つ前の知識送出位置, 1),
Mid(yourStr, 五つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 五つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 四つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 四つ前の知識送出位置, 1),
Mid(yourStr, 四つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 四つ前の知識送出位置
hitN = hitN + 1
End If
End If

```

```

End If

If 三つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 三つ前の知識送出位置, 1),
Mid(yourStr, 三つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 三つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 二つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 二つ前の知識送出位置, 1),
Mid(yourStr, 二つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 二つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 一つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 一つ前の知識送出位置, 1),
Mid(yourStr, 一つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 一つ前の知識送出位置
hitN = hitN + 1
End If
End If

ElseIf Universe.記憶遡及範囲 == 9 Then
If 九つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 九つ前の知識送出位置, 1),
Mid(yourStr, 九つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 九つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 八つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 八つ前の知識送出位置, 1),
Mid(yourStr, 八つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 八つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 七つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 七つ前の知識送出位置, 1),
Mid(yourStr, 七つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 七つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 六つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 六つ前の知識送出位置, 1),
Mid(yourStr, 六つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 六つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 五つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 五つ前の知識送出位置, 1),
Mid(yourStr, 五つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 五つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 四つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 四つ前の知識送出位置, 1),
Mid(yourStr, 四つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 四つ前の知識送出位置
hitN = hitN + 1

```

```

End If
End If

If 三つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 三つ前の知識送出位置, 1),
Mid(yourStr, 三つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 三つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 二つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 二つ前の知識送出位置, 1),
Mid(yourStr, 二つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 二つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 一つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 一つ前の知識送出位置, 1),
Mid(yourStr, 一つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 一つ前の知識送出位置
hitN = hitN + 1
End If
End If

ElseIf Universe.記憶遡及範囲 == 8 Then
If 八つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 八つ前の知識送出位置, 1),
Mid(yourStr, 八つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 八つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 七つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 七つ前の知識送出位置, 1),
Mid(yourStr, 七つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 七つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 六つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 六つ前の知識送出位置, 1),
Mid(yourStr, 六つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 六つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 五つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 五つ前の知識送出位置, 1),
Mid(yourStr, 五つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 五つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 四つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 四つ前の知識送出位置, 1),
Mid(yourStr, 四つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 四つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 三つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 三つ前の知識送出位置, 1),
Mid(yourStr, 三つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 三つ前の知識送出位置

```

```

hitN = hitN + 1
End If
End If

If 二つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 二つ前の知識送出位置, 1),
Mid(yourStr, 二つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 二つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 一つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 一つ前の知識送出位置, 1),
Mid(yourStr, 一つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 一つ前の知識送出位置
hitN = hitN + 1
End If
End If

ElseIf Universe.記憶遡及範囲 == 7 Then
If 七つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 七つ前の知識送出位置, 1),
Mid(yourStr, 七つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 七つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 六つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 六つ前の知識送出位置, 1),
Mid(yourStr, 六つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 六つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 五つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 五つ前の知識送出位置, 1),
Mid(yourStr, 五つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 五つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 四つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 四つ前の知識送出位置, 1),
Mid(yourStr, 四つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 四つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 三つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 三つ前の知識送出位置, 1),
Mid(yourStr, 三つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 三つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 二つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 二つ前の知識送出位置, 1),
Mid(yourStr, 二つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 二つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 一つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 一つ前の知識送出位置, 1),
Mid(yourStr, 一つ前の知識送出位置, 1)) <> 0 Then

```

異知識の位置(hitN) = 一つ前の知識送出位置

hitN = hitN + 1

End If

End If

ElseIf Universe.記憶遡及範囲 == 6 Then

If 六つ前の知識送出位置 > 0 Then

If StrComp(Mid(myStr, 六つ前の知識送出位置, 1),

Mid(yourStr, 六つ前の知識送出位置, 1)) <> 0 Then

異知識の位置(hitN) = 六つ前の知識送出位置

hitN = hitN + 1

End If

End If

If 五つ前の知識送出位置 > 0 Then

If StrComp(Mid(myStr, 五つ前の知識送出位置, 1),

Mid(yourStr, 五つ前の知識送出位置, 1)) <> 0 Then

異知識の位置(hitN) = 五つ前の知識送出位置

hitN = hitN + 1

End If

End If

If 四つ前の知識送出位置 > 0 Then

If StrComp(Mid(myStr, 四つ前の知識送出位置, 1),

Mid(yourStr, 四つ前の知識送出位置, 1)) <> 0 Then

異知識の位置(hitN) = 四つ前の知識送出位置

hitN = hitN + 1

End If

End If

If 三つ前の知識送出位置 > 0 Then

If StrComp(Mid(myStr, 三つ前の知識送出位置, 1),

Mid(yourStr, 三つ前の知識送出位置, 1)) <> 0 Then

異知識の位置(hitN) = 三つ前の知識送出位置

hitN = hitN + 1

End If

End If

If 二つ前の知識送出位置 > 0 Then

If StrComp(Mid(myStr, 二つ前の知識送出位置, 1),

Mid(yourStr, 二つ前の知識送出位置, 1)) <> 0 Then

異知識の位置(hitN) = 二つ前の知識送出位置

hitN = hitN + 1

End If

End If

If 一つ前の知識送出位置 > 0 Then

If StrComp(Mid(myStr, 一つ前の知識送出位置, 1),

Mid(yourStr, 一つ前の知識送出位置, 1)) <> 0 Then

異知識の位置(hitN) = 一つ前の知識送出位置

hitN = hitN + 1

End If

End If

ElseIf Universe.記憶遡及範囲 == 5 Then

If 五つ前の知識送出位置 > 0 Then

If StrComp(Mid(myStr, 五つ前の知識送出位置, 1),

Mid(yourStr, 五つ前の知識送出位置, 1)) <> 0 Then

異知識の位置(hitN) = 五つ前の知識送出位置

hitN = hitN + 1

End If

End If

If 四つ前の知識送出位置 > 0 Then

If StrComp(Mid(myStr, 四つ前の知識送出位置, 1),

Mid(yourStr, 四つ前の知識送出位置, 1)) <> 0 Then

異知識の位置(hitN) = 四つ前の知識送出位置

hitN = hitN + 1

End If

End If

If 三つ前の知識送出位置 > 0 Then



```

If StrComp(Mid(myStr, 三つ前の知識送出位置, 1),
Mid(yourStr, 三つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 三つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 二つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 二つ前の知識送出位置, 1),
Mid(yourStr, 二つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 二つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 一つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 一つ前の知識送出位置, 1),
Mid(yourStr, 一つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 一つ前の知識送出位置
hitN = hitN + 1
End If
End If

ElseIf Universe.記憶遡及範囲 == 4 Then
If 四つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 四つ前の知識送出位置, 1),
Mid(yourStr, 四つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 四つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 三つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 三つ前の知識送出位置, 1),
Mid(yourStr, 三つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 三つ前の知識送出位置
hitN = hitN + 1
End If
End If

ElseIf Universe.記憶遡及範囲 == 3 Then
If 三つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 三つ前の知識送出位置, 1),
Mid(yourStr, 三つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 三つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 二つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 二つ前の知識送出位置, 1),
Mid(yourStr, 二つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 二つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 一つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 一つ前の知識送出位置, 1),
Mid(yourStr, 一つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 一つ前の知識送出位置
hitN = hitN + 1
End If
End If

```

```

If 一つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 一つ前の知識送出位置, 1),
Mid(yourStr, 一つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 一つ前の知識送出位置
hitN = hitN + 1
End If
End If

ElseIf Universe.記憶遡及範囲 == 2 Then
If 二つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 二つ前の知識送出位置, 1),
Mid(yourStr, 二つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 二つ前の知識送出位置
hitN = hitN + 1
End If
End If

If 一つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 一つ前の知識送出位置, 1),
Mid(yourStr, 一つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 一つ前の知識送出位置
hitN = hitN + 1
End If
End If

ElseIf Universe.記憶遡及範囲 == 1 Then
If 一つ前の知識送出位置 > 0 Then
If StrComp(Mid(myStr, 一つ前の知識送出位置, 1),
Mid(yourStr, 一つ前の知識送出位置, 1)) <> 0 Then
異知識の位置(hitN) = 一つ前の知識送出位置
hitN = hitN + 1
End If
End If

Else

```

```

End If
//相互作用させる異知識の場所を, 上記で選んだ候補から
ランダムに選ぶ.
pos = CInt(hitN * Rnd())

//異知識があれば, 知識を相手に与え, 異知識がなければ,
すぐに離れるようにカウンターをセットする.

If hitN == 0 Then
//異知識が無いので, すぐに離れるように, カウンターを
ゼロに.
My.Counter = 0
return(myStr)
Else
//相互作用をする位置を代入する.
相互作用の位置 = 異知識の位置(pos)

If StrComp(Mid(myStr, 相互作用の位置, 1), Mid(yourStr,
相互作用の位置, 1)) > 0 Then
//自分の知識レベルが高い場合, 自分の知識レベルに相手
の知識レベルを合わせる.
高い知識 = Mid(myStr, 相互作用の位置, 1)
ElseIf StrComp(Mid(myStr, 相互作用の位置, 1),
Mid(yourStr, 相互作用の位置, 1)) < 0 Then
//相手の知識レベルが高い場合, 相手の知識レベルに自分
の知識レベルを合わせる.
高い知識 = CStr(CInt(Mid(myStr, 相互作用の位置, 1)) +
1)
Else
//どちらの知識レベルも同じ場合は, 自分の知識レベルを
採用する.
高い知識 = Mid(myStr, 相互作用の位置, 1)
End If

//交換した知識を元の知識の位置に戻す. 但し, 交換した

```

```

知識の後ろの知識は次の If 文で付け足す.
If 相互作用の位置 > 1 Then
//交換した知識の場所が先頭で無い場合 :
新知識 = Left(myStr, 相互作用の位置 - 1) & 高い知識
Universe.相互作用の数 = Universe.相互作用の数 + 1
Else
//交換した知識の場所が先頭の場合 :
新知識 = 高い知識
End If

//交換した知識の後ろに続く知識を元のレベルで継ぎ足す.
If Universe.PL の数 > 相互作用の位置 Then
新知識 = 新知識 & Right(myStr, (Universe.PL の数 - 相互作用の位置))
End If

//知識交換後の新しい知識を返す.
return(新知識)

End If
}

PL1 のルール :
Agt_Init{
//各 PL1 は, とりあえずランダム方向へ一歩進む.「組織」
空間の壁にぶつかったら, 元来た方向に跳ね返る.
Turn(rnd() * 360)
If forward(1) > 0 Then
Turn(180)
End If
}

Agt_Step{
//変数を定義する.

```

```

Dim MyAgt As Agt
Dim MyStaff0 As Agt
Dim 向上知識 As String
Dim PL1 たち As AgtSet
Dim MyMgr1 As Agt

//PL0 との関係性を取得する.
MyStaff0 = My.関係する PL
MyMgr1 = My.関係する PM

//PL0 と同じ位置に動く.
My.X = myStaff0 .X
My.Y = myStaff0 .Y

//全ての PL をエージェントセットとして定義する.
MakeAgtSet(PL1 たち, Universe.PROJECT.PL1)

//PM1 の送付した知識を取り込む.
If MyMgr1.送出種類の位置 >= 2 Then
//送出種類の位置が左端ではない場合.
向上知識 = CStr(9 - MyMgr1.選ばれた位置の情報レベル)
My.知識 = Left(My.知識, MyMgr1.送出種類の位置 - 1) &
向上知識 & Right(My.知識, Universe.PL の数 - MyMgr1.
送出種類の位置)
ElseIf MyMgr1.送出種類の位置 == 1 Then
//送出種類の位置が左端の場合.
向上知識 = CStr(9 - MyMgr1.選ばれた位置の情報レベル)
My.知識 = 向上知識 & Right(My.知識, Universe.PL の数
-1)
Else
//送出されたメール情報が無い場合.
My.知識 = My.知識
End If

For Each MyAgt In PL1 たち
My.知識 = My.知識

```

Next MyAgt

}

PM0 のルール :

Agt\_Init{

//各 PM0 は、とりあえずランダム方向1へ一歩進む。「組織」空間の壁にぶつかったら、元来た方向に跳ね返る.

Turn(rnd() \* 360)

If forward(1) > 0 Then

Turn(180)

End If

}

Agt\_Step{

//うろうろと動き回る。組織」空間の壁にぶつかったら、元来た方向に跳ね返る.

My.Direction = rnd() \* 360

If forward (1) > 0 Then

Turn(180)

End If

}

PM1 のルール :

Agt\_Init{

//各 PM1 は、とりあえずランダム方向へ一歩進む。「組織」空間の壁にぶつかったら、元来た方向に跳ね返る.

Turn(rnd() \* 360)

If forward(1) > 0 Then

Turn(180)

End If

}

Agt\_Step{

Dim myMgr0 As Agt

Dim k As Integer

Dim Total As Integer

Dim 送出種類の位置 As Integer

Dim a as Integer

Dim b as Double

//PM0 との関係性を取得する.

myMgr0 = My.関係する PM

//PM0 と同じ位置に動く.

My.X = myMgr0.X

My.Y = myMgr0.Y

//「メールの送出周期」で決めるステップごとにメール知識情報を PL1 に送り出してゆき、自分のメール知識情報を減らしてゆく.

//但し、送り出すメール知識情報の種類は、0 となったメール知識情報以外からランダムに選ばれる.

//メール知識情報の数は、PL の数と等しい。(各 PL の専門領域と同じ数.)

//メール知識情報が全て 0 出ないことを確認するために、各メール知識情報の種類の合計を計算する.

//合計値をリセットする.

Total = 0

//メールの送出タイミングを計る.

a = GetCountStep()

b = a / Universe.メールの送出周期 - Int(a / Universe.メールの送出周期)

//b が 0 ならば、メールを送出するタイミングである.

If b == 0 Then

//合計する.

For k = 1 To Universe.PL の数

Total = Total + CInt(Mid(My.知識, k, 1))

Next k

```

//もし合計が 1 以上の場合は、メール知識情報を減らす。
//もし合計が 0 の場合は、メール知識情報をいじらない。
If Total >= 1 Then
  ランダム選択:
  //送出するメール知識情報の種類の位置を左からランダムに選ぶ。
  My.送出種類の位置 = CInt(Universe.PL の数 * rnd()) + 1

  //選ばれた位置のメール知識情報のレベルを取り出す。
  My.選ばれた位置の情報レベル = CInt(Mid(My.知識, My.送出種類の位置, 1))

  //送出種類の位置のメール知識情報の数が 0 でなければ、一つ減らす。
  If My.選ばれた位置の情報レベル >= 1 Then
    My.選ばれた位置の情報レベル = CInt(My.選ばれた位置の情報レベル) - 1
  My.知識 = Left(My.知識, My.送出種類の位置 - 1) & My.選ばれた位置の情報レベル & Right(My.知識, Universe.PL
    の数 - My.送出種類の位置)
  //PL0 による接触情報交換の場所を優先するために、知識をメールで送った知識種類の位置を取得する。
  My.確定した送出種類の位置 = My.送出種類の位置
Else
  //送出種類の位置情報をリセットする。(この情報を PL1 が使う。)
  My.送出種類の位置 = 0
  Goto ランダム選択
End If
Else
  //メールを全て送出し終わったら、シミュレーションを終了する。
  Universe.終了判定フラグ = False
End If
End If
}

```

以上