

Title	計算機を介した共同作業の評価と共同ソフトウェア開発におけるグループウェア利用の検討
Author(s)	村越, 広亨; 海谷, 治彦; 落水, 浩一郎
Citation	Research report (School of Information Science, Japan Advanced Institute of Science and Technology), IS-RR-97-0022S: 1-39
Issue Date	1997-05-20
Type	Technical Report
Text version	publisher
URL	http://hdl.handle.net/10119/8431
Rights	
Description	リサーチレポート (北陸先端科学技術大学院大学情報科学研究科)

計算機を介した共同作業の評価と
共同ソフトウェア開発におけるグループウェア利用の検討

村越広享 海谷治彦 落水浩一郎

1997年5月20日

IS-RR-97-0022S

北陸先端科学技術大学院大学

情報科学研究科

〒923-12 石川県能美郡辰口町旭台1-1

{murakosi,kaiya,ochimizu}@jaist.ac.jp

©Hiroyuki Murakoshi,1997

ISSN 0918-7553

もくじ

1	はじめに	2
2	共同作業を分類するための特徴	3
2.1	タスクの種類	3
2.2	グループのメンバー	4
2.3	作業環境	5
3	計算機を介した共同作業の評価	6
3.1	作業方法	6
3.2	コミュニケーション	7
3.2.1	コミュニケーションの評価手段	7
3.2.2	メディアの選択基準	9
3.2.3	作業者間の認識	11
3.2.4	コミュニケーションプロセス	15
3.3	作業や会話の調整	17
3.4	作業効率と成果物の品質	18
4	共同ソフトウェア開発	20
4.1	タスクの種類	20
4.1.1	Requirements Elicitation(要求獲得)	21
4.1.2	Specification Development(仕様の決定)	21
4.1.3	仕様のレビュー	21
4.1.4	Coding(プログラミング)	22
4.1.5	Code Inspection	22
4.1.6	Debug and Test	23
4.1.7	Maintanance	23
4.1.8	各フェーズのタスクタイプ	23
4.2	グループのメンバー	23
4.3	作業環境	24
5	共同ソフトウェア開発支援	25
5.1	共同ソフトウェア開発のケーススタディ	25
5.2	支援方法, 支援ツール	25
5.3	共同ソフトウェア開発ツールの評価	29
6	まとめ	31
6.1	実験方法/評価手段の考察	31
6.2	自在システムにおけるコミュニケーション支援の評価	33

1 はじめに

計算機を用いて、作業効率の改善や、生産物の品質の向上の支援を効果的に行うためには、計算機システムそのものの研究と、支援対象である作業の理解の2つが不可欠である。今日、特に人間が共同で行う作業に関する支援が注目を集めている。そのような研究分野はCSCW(Computer Supported Cooperative Work)と呼ばれており、その名の通り、計算機や通信技術などの計算機支援(CS)の研究と、言語や認知などの観点から共同作業の理解(CW)をする研究を柱としている。また、グループや組織の作業のためのアプリケーション・ソフトウェアは、グループウェアと呼ばれており、既に数多くのシステムが開発されている[1]。CSCWでは、支援対象である作業をモデルや法則として記述する場合が多い。支援を念頭においたモデル化を行う場合、作業に影響を与えるような特性と、その特性間の関係に注目すべきである。そのようなモデルが記述できてはじめて、作業に適切な計算機支援を構築したり選択したりすることができると思われる。モデル構築のためには、実作業の観察や実験的作業の分析などの心理学的な手法が用いられる場合が多い[2]。

しかし、現状では、適切な支援が行われていない場合もある。例えば、ソフトウェア開発では、多くが共同作業で行なわれているが、そこではグループウェアが効果的に利用されていないことが報告されている[3]。これは、グループウェアを通して適用された計算機技術や作業モデルが、支援対象である作業に適合していなかったことが原因の一つと考えられる。

本報告では、特に対象をソフトウェア開発に絞り、そこでのグループウェアを評価する研究についてのサーベイを通して、適切なグループウェアの構築/選択の方法について考察する。2節では、共同作業を分類するための特性を紹介する。計算機による支援は、これらの特性を考慮して決定されるべきである。グループウェアの評価は、結局のところ、グループウェアを適用する作業に対して、そのグループウェアが適切か否かによって決定される。よって、具体的な評価はグループウェアそのものの特性だけではなく、グループウェアを適用した作業過程や、作成された生産物などを通しても行われる。そこで、3節では、現在までに行なわれている共同作業の評価についての研究をまとめる。4節では、ソフトウェアの開発フェーズ毎の共同作業の特徴について整理する。そして、5節では、実際にグループウェアを利用した共同ソフトウェア開発の研究事例や、その評価方法についてまとめる。最後に、上述した考察をもとに、我々が行なっている共同作業の評価研究について議論する。

2 共同作業を分類するための特徴

計算機による共同作業の支援は、作業の特徴に合わせて行われるべきである。本節では、文献[2]で着目されている作業の3つの特徴である、タスクの種類、グループのメンバーとその構成、作業環境の種類について概説する。これらの特徴は、グループのインタラクションやパフォーマンスに大きな影響を与える要素である[4]。よって、それらの向上や改善を目標とした計算機支援を評価する場合にも、重要な要素となると考えられる。

2.1 タスクの種類

タスクの違いに応じて、その遂行方法などが異なるかどうかを系統的に調査した研究は、ほとんどみられなかった。もし、このような関係を示すようなタスクの分類ができるのなら、その分類は、タスクの遂行を支援するシステムの選択に貢献するはずである。

McGrath[5]は、集団によるタスクの遂行に関する以下のような問題の考察を行っている。

1. 集団作業が個人の作業よりも効果的であるか。
2. どんな種類のタスクが、集団で遂行するのに向いているのか。
3. タスクの遂行過程のどのような側面が、集団作業の効果に影響しているか。

その考察をもとに彼は以下に示すようなタスクに関する分類を行った。その分類は以下のような条件を満たしている。

1. 相互に排他的である(タスクに適合するカテゴリはただひとつ)。
2. 集合の余分がない(すべてのタスクがカテゴリに分類できる)。
3. お互いが論理的に関係している。
4. タスク間の相違や関係を指摘しているといった点で役にたつ。

タスク分類のスキーマは以下のようになっている。タスクは2つのサブタイプをもった、4つの次元に分類することができる(Task Cricumplex: 図1参照)。

1 次元: 考えや計画を生成。

- Planning tasks(タイプ1): 計画を生成する。
- Creative tasks(タイプ2): 考えを生成する。たとえば、ブレインストーミングなど。

2 次元: 正しい解や好ましい解を選択。

- Intellectual tasks(タイプ3): 集団の意見の一致によって、正しい答えを見つける。このタスクにおいて、直観的に、決まった(強制的な)解をもっている場合には、「真実が勝る」といった原理で作業が進む。また、明白な解がない問題に対しては、解の正当性を論証することは難しい。ここでの作業は、「その解を支持する真実が勝る」といった原理で進められる。ほとんど明白な解がないような問題に対しては、「多くの支持を得た真実が勝る」といった原理で作業が進められる。
- Decision-making tasks(タイプ4): 集団の意見の一致によって、より好ましく、同意できる答えを見つける。つまり、正しい答えを持たない問題に関して集団のコンセンサスを形成すること。このタイプの作業の欠点は、

- 集団は、集団が持っている技術や知識を十分にそして効果的に利用していないかもしれない。
- 何人かのメンバーは、他人より多くの影響をもったり、知識をもったりするかもしれない。
- 良い決定をするというよりも、すぐに同意をするためのプレッシャーがあるかもしれない。
- メンバー間の知識の多様性が見解と価値観の多様性を発生させる。それは、単一の決定を行なうことを困難にさせているかもしれない。

3 次元: 意見や興味の衝突の交渉。

- Cognitive conflict tasks(タイプ5): 集団内の見解の衝突を解決すること。
- Mixed-motive tasks(タイプ6): 興味の衝突を解決する。

4 次元: 相手や外部のパフォーマンスの基準との競争。

- Contests/battles/competitive tasks(タイプ7): 他者と競争し、他者との衝突を解決すること。
- Performance/psycho-motor tasks(タイプ8): 優れた外部の基準と競争すること。

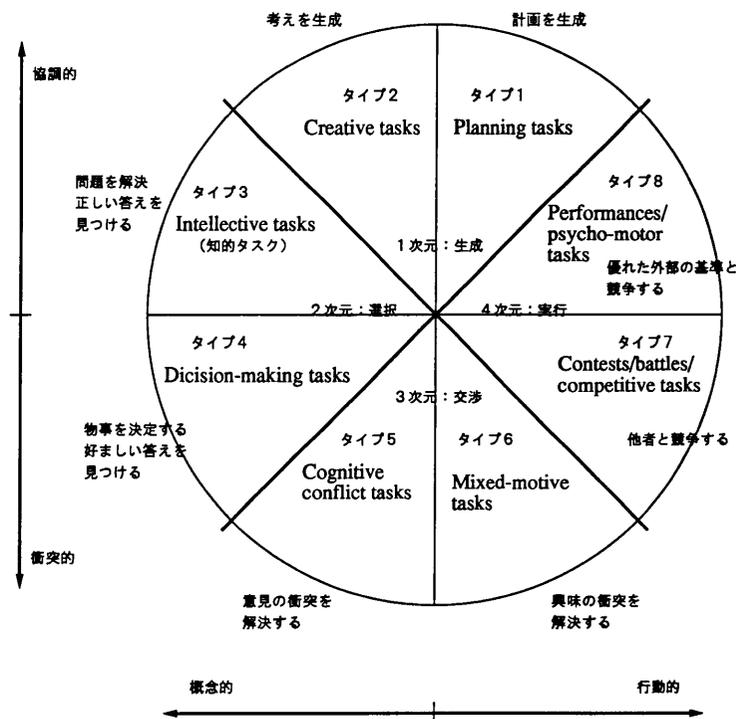


図 1: 集団作業のタスクの分類

2.2 グループのメンバー

文献 [2] によると、グループのメンバーを以下のような特徴から分類することで、適切な支援方法を選択することが可能だとしている。

- グループを構成する個人の特徴: 様々な専門知識や才能, 心構え, パーソナリティ, リーダーなどの役割。

- メンバー間のインタラクションの特徴: どれくらいの間, お互いが知り合いなのか. たとえば, リーダーなどの役割を適切な参加者に割り当てることは, 集団作業の効果的な遂行に貢献すると考えられる. また, これらの特徴は, グループの構成人数によって変化する. たとえば, 小規模なグループでは, 作業の各段階毎にリーダーが変化する場合がある. 計算機の支援を考える際には, そのような研究を進め, CSCW に必要な機能の提案を再考すべきである.

2.3 作業環境

グループが作業を遂行する状況を作業環境と呼ぶことにする. 例えば, 同室で作業をする状況もあれば, 電話等を用いて異なる場所で作業する状況などもある. 共同作業を行なう環境は, 作業方法やコミュニケーション方法などに影響を与えると予想できる. 例えば, 地理的に制限された作業者は, 電子会議システム等を利用せずに共同作業を行なう場合, 相手の様子を知ることができないといった制限が加えられてしまうために, 電話等を用いて相手の作業を中断させてしまう可能性がある.

Johansen[6]¹によると, 時間と場所の次元によってグループウェアを4つに分類した. 以下にそれを示す.

- 同じ時間, 同じ場所: 対面会議
- 同じ時間, 違う場所: 遠隔電子会議
- 違う時間, 違う場所: 共同執筆
- 違う時間, 同じ場所: 同一プロジェクトのシフト作業 (病院・工場)

石井 [7] も同様に, 時間と空間の2つの次元を用いて, グループウェアの以下のように分類した.

- 対面・リアルタイム型 (対面同期型): 電子会議システム
- 分散・リアルタイム型 (分散同期型): TV 会議システム
- 分散・蓄積・非同期型 (分散非同期型): 電子メール

Johansen と石井の分類の相違は, 場所の特性のパラメータの違いと考えられる. つまり Johansen は, 同じ場所で働く/異なった場所で働くといったパラメータを考えているのに対して, 石井は, 対面/遠隔といったパラメータを考えている. 石井の分類では, 対面非同期型グループウェアは存在しない.

¹Johansen は, 上述した分類以外にもグループウェアを, 会議の規模 (参加人数) の大小から分類したり,

- 対面会議を支援する.
- 電子会議を支援する.
- 会議の間の作業を支援する.

といったようにも分類している.

3 計算機を介した共同作業の評価

共同作業が効果的に支援されているのを明白にするためには、システムの評価を行なう必要がある。特に定量的に評価することができれば、他のシステムとの比較が可能となり、より効果的なシステムを選択することが可能となる。あるいは、より効果的なシステムに改良することが可能となる。よって、ツールを適切に評価する手段を分析する必要があると考えられる。この節では、共同作業の評価に関する研究をサーベイし、評価手段について検討する。共同作業の評価研究を以下の観点からまとめる。

- 作業方法
- コミュニケーション
- 作業や会話の調整(コーディネート)
- 作業効率や成果物

3.1 作業方法

共同作業を行なうためには、情報の獲得や意見の創造、意見の評価等の作業を行なう必要がある。これらの作業方法は、計算機支援の導入によって変化すると考えられる。よって、計算機導入が作業方法に与える影響に関する研究をまとめる。

共同執筆作業の評価研究が数多く行なわれている。例えば文献 [8] では、分散共同執筆の3つのケーススタディを紹介している。共同執筆の過程は、変化する状況に絶えず適応しながら展開していく。その中で各々の執筆者は限られた情報を提供・利用する。その新しい状況に対して適切な決定を行なうために、それらの情報を便宜的に利用する戦略を *informed opportunism* と呼ぶ。この *informed opportunism* を観測から獲得することが、分散した人々による共同執筆の支援方法を設計する際に、何が関係して、何が必要な情報か、行なわれている事が何であるかといった概念を考察する手がかりとなっている。また文献 [9] では、共同執筆作業に役立つインタラクションパラメータを定義し、そのパラメータを選択できるような共同執筆支援ツール、PREP Editorを開発している。そしてそのパラメータについての有用性について議論している。インタラクションパラメータとして、

- 共同執筆作業グループのタイプ
- 執筆のスタイル
- 共同執筆中の役割
- 共同執筆に不可欠なものの変化

を定義している。さらに分散執筆に適用するために、

- 結果の共有
- タスクの管理

といったパラメータを定義している。

また共同ソフトウェア設計作業の評価研究も行なわれている。例えば文献 [10] では、グループがどのように活動し、何がうまくいっていて、何がうまくいっていないのかを調べるために共同作業の実験を行なった。その結果は作業改善の効果を測定するための基礎となるだけでなく、グループのメンバーがどのような道具を必要としているかを示

すとしている。10の共同ソフトウェア開発の初期の作業の分析では、作業中の活動の領域を分類し、その分類に要した作業時間や選択肢に関係する議論に着目している。分析結果であるが、作業の全体の40%が設計の選択肢とそれに関する評価の議論に費やされていた。また30%が自分たちの作業経過とまとめの評価に、20%が作業の調整に費やされていた。そして、ほとんどの議題に対して1つ以上の選択が示され議論されているにも関わらず、幅広く議論されているものはほとんどなく、3分の1は明確に評価されていないといった結果が得られた。ただこの研究では、成果物に対する評価は行なわれていない。これは、プロジェクトが異なる作業同士の生産物を比較するのは、非常に困難であるからだと考えられる。

またShrEditといった共有エディタを用いることにより、作業者がシステム設計作業に集中することができ、設計を探索する範囲が狭くなったので、効率的に作業を進めることが出来たといった文献がある[11]。これは確かに興味深い結果であるが、その効果が得られた理由については、明確にされていない。

同期型グループウェアの設計に着目し、それに必要な機能を分析するための枠組を提案した文献[12]がある。様々な状況下で、人間の活動とシステムの支援がうまく適合するように、グループウェアの機能を組み合わせるといった考え方がとても興味深い。具体的には、同期作業を4つの活動領域に分割し、3つの連結概念である、linking, awareness, undoを用いて、同期作業グループウェアの機能枠組に適応させ、それを用いた評価方法を提案している。

3.2 コミュニケーション

グループウェアの導入は、共同作業の中でも特にコミュニケーションに大きな影響を与える。そこで本節では、コミュニケーションに関する評価研究を以下の観点からまとめる。

- メディアの選択基準
- 作業者間の認識
- コミュニケーションプロセス

3.2.1 コミュニケーションの評価手段

対面のコミュニケーションと、計算機を介したコミュニケーションとの比較を行なっている社会心理学や人間的要因に関する研究の長い歴史がある。それらの研究では、コミュニケーションの方法における優位な差異をみつけることができなかった[13][14]。具体的に文献[13]では、以下に示す3つの問題から、研究を進めている。

1. 問題解決において人々が情報交換するとき、お互いがどのようにしてコミュニケーションを行なうのか。
2. コミュニケーションは、人々が会話する道具によってどのように影響されるのか。
3. メディア以外の他のどんな変数が、コミュニケーションに影響を与えるのか。

またその文献では、コミュニケーションパターンについて、以下の4つの問題を示している。

1. コミュニケーションパターンは、様々な国の人々の間でどのように変化するのか。

2. コミュニケーションパターンは、コミュニケーションの目的に応じて、どのように変化するのか。
3. コミュニケーションを行なう人の数が増加するならば、コミュニケーションパターンに何が発生するのか。
4. 自然なヒューマンコミュニケーションを支配する規則は何であろうか(文法, シンタクティック, セマンティック). コミュニケーションの文法がきれいでも、コミュニケーションが成立し成果物が生まれるのは、何故だろうか。

Williams は文献 [14] において、様々なメディアの効果に関する研究を行なっている。実験では、正しい解を発見するといったタスク²を採用した。結果として、

- Text 型のメディアは、Voice 型のメディアとは異なる。Voice 型のメディアの方が、より多くのメッセージの利用しているが、より早く解を見つけることができる。
- Text 型のメディアの間には、ほとんど相違がない。また、電話のように音声のみと対面の場合を比べた場合、環境にかなりの相違があるが、Voice 型のメディアの間にも相違がほとんどない。
- 実験で用いたタスクでは、非言語的な合図(身振り等)がほとんど重要ではない。

といったことを導いている。

2.1節におけるタイプ3の Intellective tasks の多くが、メディアに依存していないことを示した文献 [15] がある。具体的に Gale は、3つの作業環境を以下のように設定した。

1. データの共有
2. データの共有 + 声
3. データの共有 + 声 + 映像

後述の環境ほど、高いバンド幅の通信経路が必要である。この3つの作業環境の間には、生産物の品質、タスクを遂行するまでにかかった時間に差異がみられなかったとしている。しかし、高いバンド幅のコミュニケーションは、社会的、インフォーマルなコミュニケーションに効果があったことを示している。

計算機を介したコミュニケーションを評価手段の一つとして、会話分析が用いられることが多い。文献 [16][17] によると、会話分析とは、社会相互作用に関する分析の一般的なアプローチである。このアプローチは、発話を実際に話されたコンテキストから単独で抜きだし、分析者が主観的に発話行為 [18] を確定し分析していくのではなく、会話のコンテキスト(会話のシーケンス構造上の位置付け)によって会話中の文章や発話の内容を理解し、会話の構造を分析する方法である。実際の会話構造から考えると、会話は、(あいさつ, あいさつの答え), (提案, 了承), (質問, 解答) といったような隣接した発話行為から構成されていることがわかる。このような隣接した発話行為のペアは、隣接ペアと呼ばれている。隣接ペアは、第一ペアと第二ペアから構成されている。この第二ペアは、構造的に優先的されるものを非優先的なものに分類できる。もし非優先的な第二ペアが発生する場合、2つの発話だけでは完結せず、複雑な会話が展開することになる。

言語的コミュニケーションにおける基本単位は、記号や語、文といったものではなく、発話行為における記号、語、文の発生である。発話行為とは、「陳述」、「命令」、「質問」、「約束」などの行為を遂行することである。記号、語、文のトークンを一定の状況下で

²ここでの作業は2.1節の分類を参考にすると、タイプ3の Intellective tasks となる。

産出することないし発話することは、なんらかの言語行為そのものである。つまり発話行為は、言語的コミュニケーションの基本的な単位あるいは最小の単位としている [18]。言語を利用することは、発話行為を遂行することである。

3.2.2 メディアの選択基準

人々は通常、対面や電話といったように、一つ以上のコミュニケーションメディアを利用しており、適材適所でメディアを選択している。文献 [1] では、集団の個々の振るまいを一つの集団の振るまいとみなし、タスクの種類や作業状況に応じたメディアの選択を行なうべきだとしている。しかし、そのようなメディアの選択基準はタスクの種類や作業状況以外に、新しいメディアの導入によって変化することが予想される。ここで、以下に示すメディアの特徴を示す理論に基づいて、メディアの選択基準について考察する。

- *media richness*(メディアリッチネス)[19]
- *critical mass*(クリティカルマス) [20]

これらの理論では、メディアの選択やメディアの使用は、メディアの特徴によって決定されるとしている。

文献 [19] では、組織内の様々な不確かさや曖昧さを減少させるための情報のやりとり
に焦点をあて、組織構造に依存する効果的な情報のやりとりを提案している。不確かさと曖昧さに関して、以下のように定義されている。

不確かさ: 情報の欠如を意味している。つまり情報が増加するならば、不確かさは減少する。

曖昧さ: 組織内における多様で衝突的な解釈の存在、つまり多義性 (ambiguity) を意味している。高い曖昧さは、混乱と理解の欠如を意味しており、どんな質問をして良いのかわからないし、質問をしても、相手からの次にやってくる返答のポイントがよくわからないことが起こる。

曖昧さを減少させるためには、組織内に大量のデータを単純に提供するよりも、議論や説明を効果的に行なう必要がある。そのためには、rich information を容易に利用できることが重要となってくる。この rich information は、“ある時間間隔内で理解を変更するための情報の可能性” と定義されている。豊かな rich information とは、ある時間内で理解を変更するために、情報同士の関係を理解し、曖昧な記述を明らかにすることができる情報のことをいう。反対に、乏しい rich information がは、理解するために長い時間がかかり、様々な見通しを克服できないような情報のことをいう。つまり richness は、コミュニケーションによる学習可能性を示しているといえるとしている。コミュニケーションメディアは、rich information の実行可能性によって異なる。この豊かさを、返答の即座性、コミュニケーション手段の数、利用できるチャンネル数、人格、言語の多様性などから決定することによって、豊かなメディアの順番を以下に提案している。

1. 対面
2. 電話
3. 手紙やメモなどといった個人的文書
4. 非個人的な文書
5. 統計の文書

対面の場合がもっとも豊かなメディアであるといえるのは、即座の返答によって、自分が相手に伝えたメッセージの解釈を確かめることが可能だからである。さらに、身ぶりや声の調子といったような多様なコミュニケーション手段をもち、会話内容を母国語で行なうことができるからである。豊かなメディアは、複雑で主観的な会話を行なう可能性を提供することにより、あいまいさを容易に減少させることが可能にする。このように定義されるメディアの豊かさを media richness という。

media richness と同様に、コミュニケーションメディアの豊かさを定義し、コミュニケーションを評価した研究がある [21]。具体的には、メディアの豊かさを以下の2つの特性から定義している。

interactivity: 返答の即座性と適確性

expressiveness: 道具や言語の多様性、人間感情の移入性

共同執筆支援ツールの効果的なコミュニケーションを探求するために、expressiveness が豊かな Voice 型のコミュニケーションと、乏しい Text 型のコミュニケーションの比較を定量的に行なっている。実験は、文書修正の注釈づけといった共同作業で、その作業は3種類のタスクに分類できた。複雑で曖昧さが多いタスクの場合、Voice 型のコミュニケーションの方が効果的であり、タスクの種類毎に効果的なコミュニケーションメディアが異なるといった結果を示している。しかしながら、interactivity の豊かさの相違によるコミュニケーションへの影響については議論されていない。

critical mass とは、ツールを利用するのに必要な人数のことを示し、ツールを評価するための尺度である。しかし、ツールを利用するユーザー側の要求の相違から、ツールの利用人数に差異が生まれてしまう。例えば、以下に示すようなツール利用が考えられる。

- 高い利便性のしきい値を持っているので、あるツール開発の初期バージョンを利用せず、成熟段階でのバージョンを利用する。
- また、自分に関係する人が多く利用している方が好ましいので、関係のない人が多く利用しているツールを使わない。

グループウェアが効果的に運用されるためには、利用人数がある値まで揃わなければならないといった critical mass の壁がある [20]。コミュニケーション技術の開発の場合、多くの人とのコミュニケーションを可能にする必要があり、critical mass は重要である。よって、グループウェアを効果的に利用するためには、critical mass の壁が存在し、その壁を破るために様々な考察が必要であると考えられる。一つの方法として、グループウェアを利用する個人の制約を少しでも取り除くといったこと考えている。

文献 [22] では、

- 過去の共同作業に関する論文 (55 本) の分析結果
- 共同執筆作業において、対面のコミュニケーションと電子メールの比較を行なった実験結果
- 文書の注釈作業のコミュニケーションメディアとして、Voice と Text を比較した実験結果

から、以下の仮説について検証している。

- 文書作成といったタスクの曖昧さが増加すると、表現が豊かで双方向性に富んだ rich communication を支援するような道具が利用される。

- 富んだ rich communication を支援するような道具が利用されるならば、曖昧なタスクは、より簡単に良い成果物を得ることが可能となる。

またタスクの特徴の測定が、タスクと計算機技術との関係を理解するのに必要であることが示されている。つまりタスクの種類が、利用すべき計算機技術に大きな影響を及ぼしていることを示している。さらに、どの計算機技術を利用するかを決定するために、media richness の構成要素である、双方向性と表現性の区別についても考慮に入れることが重要であるとしている。

世の中には様々なメディアが開発されているが、人間は効果的なメディアを選択し、その他のメディアは使用されない。Kraut らは、人間が新しい科学技術を採用し利用し続ける理由には少なくとも、利用性(コストの良否)と社会への影響(social influence)といった観点から説明できるとしている[23]。これらの考え方は、Jovanis らの文献[24]が基礎となっている。Jovanis らは、組織やユーザー、コミュニケーションタスク、様々なコストパターンの特徴を認識するためのモデルを提案し、メディアを選択する基準を導入している。

過去に計算機を介したコミュニケーションの研究が行なわれているにも関わらず、メディアが与える人間の振るまいや慣習への影響についての理解はほとんどなされていない。Jovanis らは以下のように、それらの研究の欠点を指摘している。

- 様々な分野からの結果を試したり、統合したりするアプローチの欠如
- 新しいメディアの効果を分析するのに適応したフレームワークの欠如
- 新しいメディアを選択するための効果を説明、テストするための定量分析ツールの欠如

さらに、そのような欠点の解決方法として、以下の提案している。

- 新しいコミュニケーション支援技術の効果に関する様々な研究から得られる発見の融合。
- 新しい科学技術の影響の探求をガイドするための概念的なフレームワークの開発。
- 定量的なデータ集合や分析技術を利用して、フレームワークの特別な構成要素や、個々のメディアの選択の修正。

また、個人が様々なコミュニケーションメディアに認識するものを定量的に評価した。個人というのは、適切な特性をメディアの選択の基準として採用しており、それは、適切なメディアの選択の基準を決定する指標になると述べている。また、組織の中のメディア選択モデルを構築するには、組織の背景に基づいた個人の選択に適合するモデルの構造が基本であるとしている。これらの考え方は、メディアの選択基準を決定するためのフレームワークに役立つと考えられる。しかし、選択の基準となる評価パラメータについての良否の検証がなされず、さらなる考察が必要だと考えられる。

3.2.3 作業者間の認識

作業者同士がコミュニケーションを行なう場合、様々な状況から、お互いの認識が一致しないことがある。そのような認識の不一致は、効果的なコミュニケーションを妨げることが予想される。以下に、作業者間の認識の一致の度合の代替特性、およびそれを用いた評価研究について説明する。

common ground 複数の人間が共同で問題を解決するとき、各々の perspective や専門知識を利用する。perspective とは、

- 目的
- 背景知識
- 仮説

などのことをいい、各々の作業で異なっている。さらに、専門知識 (プログラミング言語の熟知度など) も各々の作業間で差があると考えられる。そのような perspective や専門知識の多様性は、問題解決には利点となるが、労力の調整には困難となる。作業が、他の参加者の perspective が自分の perspective とどのように異なっているのかに関する知識を獲得できるならば、様々な perspective が利用可能であると指摘されている [25][26][27]。Clark による談話の contribution theory は、人々がそのような知識を達成し、維持するためのメカニズムの基礎を提供している [28]。この理論の中で使用されている *common ground* とは、会話における参加者の相互知識 (mutual knowledge)、信念、仮定のことを意味している。会話中における *common ground* は、各々の参加者が現在の目的のために、他の参加者が自分の発話を十分に理解できるように試み、規則的な方法で更新するとされている。Clark によると、*common ground* の獲得に必要な作業は、利用できる環境の制約の相違によって変化するとしている。

文献 [29] では、Clark がたてた、「一緒にいるとか、目に見えるといったような制約がない場合、*common ground* を達成することが難しい。」といった予想を確かめるために、実験を行なっている。具体的には、銀行のレイアウトを決定する問題をタスクとして、離れた場所にいる被験者のペアに、Text ベースの同期型コミュニケーションを行なわせている。また作業中は、自分のプライベート空間に決定事項とそれに関する議論を記述することを可能としている。結果として、Text ベースのコミュニケーションでは、*common ground* の達成が困難であることを導いている。*common ground* の達成の欠如は、作業終了後に、各々の被験者に作業中に決定した決定事項とそれに関する議論を思いだしてもらい、その結果の相違によって測定している。さらに、共有作業空間を持つシステムを利用した場合、その困難さを減少させることが可能になるといった結果を導いている。具体的には、特に決定事項に関する議論の相違が少なくなるとしている (反対に決定事項数に関しては、あまり相違がなかったとしている)。この理由であるが、共有作業空間を持たない作業の場合、決定事項について、新たに自分の個人作業空間に記述しなければならないといった余分な作業を行なう必要があり、また決定事項の方が決定するための議論の方より重要であると考えられるからだと予想している。また、個人の作業空間に記述される相手と異なった議論が、決定事項よりも多いということは、人間は相手の強い否定の言葉を認識しないかぎり、自分の意見を認めてしまうためであるとし、グループ活動の調整の必要性を提案している。

他に *common ground* に関する文献には [30] がある。著者の Krauss らは、コミュニケーションを効果的に行なうためには、相互知識³(mutual knowledge) に関する問題、つまり次に何を言うかを決めるために、コミュニケーションの相手が何を知っていて、何を知らないかに関する情報を得なければならないといった仮定⁴から研究を進めている。

³著者らは、この相互知識と *common ground* を同一視している。

⁴相互知識問題と呼ばれている。

まず共同作業で発生するコミュニケーションのうちで、行為者が情報を伝えるために意図的に行なったものに焦点をあて、common ground によって、それらがどのような効果を得ているかをまとめている。また、計算機を介したコミュニケーションで common ground を得るためには、どのような労力が必要であり、どの方法が最も効果的であるかを分析し考察している。

コヒーレンス Reichman は、コミュニケーションのコヒーレンス⁵の達成について言及している [31]。その達成のためには、すべての会話を行なっている者がコンテキストを共有する必要がある。Reichman はコヒーレンスの達成のために、会話情報のなかから余分な情報を新たに処理することによって、その問題を解決することを考察している。また会話のコヒーレンスを維持するプロセスを明らかにすることを目標としており、その目標を達成するために、“context space” と呼ばれる構成物を分析対象としている。この context space を形成するのは、ある問題や話題に言及している発話のグループである。つまり、発話が階層的に関連している context space に分解され、構成されたものである。会話が進められると、各々の発話を行なっている者は、会話の context space の構造や、現在の議論のトピック、焦点が当てられている項目のリストといったものを含んだ会話モデルを構築する。会話のコヒーレンスは、参加者の間の各々の会話モデルの間の衝突 (conflict) のないことに依存している。その衝突をふせぐには、聞き手が話し手の会話モデルを認識することを可能にする規則が必要である。Reichman はその規則を、階層的にそれぞれに関連している context space から明らかにしている。

McCarthy らは文献 [32] の中で、Text ベースのコミュニケーションにおけるコヒーレンスが、どのようにして達成され、維持されるのかについての分析を行なっている。具体的には、survival task と joint submission task といった、共同に作業者が物事を決定する作業を Text ベースのコミュニケーションを用いて行ない、分析をしている。結果として、複数の異なった話題が並行に進められていることがわかった。この原因として、Voice ベースの会話は、一時的なもので話題の回顧はかなりの記憶力が必要となるので、単一的话题を進める傾向にあるが、反対に Text ベースの場合は、コミュニケーションが記録をしやすく、並行に話題をすすめるための環境を提供しやすいことが予想された。しかし、Text ベースのコミュニケーションに特徴的な「読み書きによる遅れ」が、コミュニケーションの非同期性を増加させていることが原因とも予想され、作業者が並行に話題を進めるかについては説明できないとしている。

さらに Text ベースのコミュニケーションでは、face-to-face の会議のようなターンテキングが利用できないので、会話のコヒーレンスの達成の手段として、以下のことを提案している。

- addressing(明確な主張): ある話題の返答を行なうときに、明示的に返答する話題を明示すること。ただし、隣接ペアについては例外である。
- sequential organization: 並行の話題に対して、それぞれに導入された順番に返答すること。その返答の詳細は、次の話題の返答の間に行なう。

⁵和訳は「一貫性」になる。文献中の意味は、Text 型による相手からのメッセージと、そのメッセージのコンテキストを結びつけること。

- message compression: 複数の話題を並行に進めないようにすること。これは、メッセージを短くしたり、メッセージを情報パケットに分解することによって達成できる。

また結論として、以下のことを考察している。

- 効果的なコミュニケーションを行なうためには、メディアに必要な会話のスタイルと構造を適用することが必要である。
- 対面の会議やビデオ電子会議システムと比べて、Text ベースの会議は、限られたコミュニケーションチャンネルしか持たないが、効果的なコミュニケーションへの障害とはならない。

さらに今後の課題として、以下に示すものが、複数の話題が並行に進むことに影響を及ぼすかどうかの分析が必要だとしている。

- グループのサイズ
- タスクの種類

共同作業を評価する方法の一つとして、タスクの遂行を測定することが考えられる。例えば、どのくらい良く、どのくらいすばやく作業の遂行が終了したかを測定することである。しかし実際には、一人の作業者が一時に一つだけのタスクに従事することはまれであるため、一つのタスクの遂行のみを調査するだけでは不十分である。そこで McCarthy らはタスクの遂行がコミュニケーションの利用のしやすさの変化の総計に依存するということから、コミュニケーションの評価をすることを推奨している [33]。しかし実際は、様々な形態のコミュニケーションを比較した実験的研究が盛んに行なわれ、コミュニケーションの効果などを測定しているが、あまりうまくいっていない。McCarthy らは、それらが役立つものになるためには、以下のことが必要だとしている。

1. 過去に行なわれている対面と計算機を介したコミュニケーションの比較実験では、得られた結果が実験タスクに非常に依存しているので、適切な実験タスクを利用すること。
2. 適切な独立変数を測定すること。

McCarthy らはそのような点に注意して、Text ベースの会議システムの 3 つの状況を比較することを行なっている。成果物の相違は見られなかったが、多くのコミュニケーションプロセスの独立変数には差異がみられた。それらの変数として、

- shared recall⁶
- 同一話題の参照の距離

を導入している。評価対象としたのは、Text ベースの会議システム “Conferencer” であった。これは、

- 共有作業空間の使用
- 会話の構造

の違いを分析することが可能であった。そこで、この Conferencer を用いて、

- 共有作業空間がある/ない
- 過去のコミュニケーションの記録を参照できる/できない

⁶実験後に、作業者が共通に記憶している話題(成果物)のこと。

といった違いのある実験環境を設定した。また、被験者のペアの設定方法について議論している。環境が3つあって、それぞれの環境で実験を行なうペアを3つとした場合、以下に示すものが考えられるとしている。McCarthyらは、設定1を利用して実験を行なっている。

設定1: 各々の被験者が一つの環境で作業を行なう。

- 環境1: ペア A, B, C
- 環境1: ペア D, E, F
- 環境1: ペア G, H, I

設定2: 文献[34]で採用されていて、各々のペアがすべての環境で作業を行なう。利点としては、個人差の影響を排除できるが、欠点として、環境を経験したことによる影響が考えられる。

- 環境1: ペア A, B, C
- 環境2: ペア A, B, C
- 環境3: ペア A, B, C

設定3: 各々の被験者がすべての環境で作業を行なうが、各々の環境でのパートナーが異なる。欠点として、新しいパートナーと組んで作業を行なう影響が考えられる。

- 環境1: (A1,A2), (B1,B2), (C1,C2)
- 環境2: (A1,B1), (C1,B2), (A2,C2)
- 環境3: (A1,C1), (C2,B2), (C1,A2)

コミュニケーションの効果は、common groundの達成といった観点から評価している。このcommon groundを計る尺度として、決定解とそれに関する議論の数やお互いに相違した決定解とそれに関する議論の数を導入し、分析を行なっている。さらに会話の内容の分析として、

- 1, 2人称代名詞(私, 私たち, あなた)
- 指示代名詞(それ, あれ, それら)

をとりあげ、分析を行なっている。その結果、共有作業空間を使用した方が1, 2人称代名詞の使用が多いことがわかり、円滑で社会的なコミュニケーションが行なわれていることを予想している。指示代名詞の場合、過去の会話の履歴がある/ないでは差異はみられなかった。それについては、各々の環境で会話の隣接ペアが結び付いているからだとしている。また会話の構造の相違を分析するために、同じ話題の参照距離を測定している。その参照距離は、同じ話題の間で行なわれた発話のターン数として測定している。結果として、過去の会話の履歴があった環境では、参照距離が長いことが得られている。

3.2.4 コミュニケーションプロセス

文献[35]では、コミュニケーションプロセスを評価するための方法を提案し、計算機を介したコミュニケーションの評価について考察している。具体的には、コミュニケーションの特徴を分析するために、コミュニケーションプロセス変数を定義し、その変数の評価を行なっている。以下にその特徴をまとめる。

- 人称代名詞(私, 私たち, あなた)の使用回数: 1人称や2人称の使用は, 間接的な質問において使用され, 直接的な質問よりも礼儀正しいといった感じがある. これらは, インタラクションの社会的なコンテキストや社会的な慣習, 関係に関連している.
- 指示代名詞(それ, あれ, それら, 彼, 彼女)の使用回数: 指示代名詞は上述されている名詞句を指す. その使用は, 名詞句の繰り返しよりも, メッセージの長さという観点から考えると効率がよい. そして, それを使用するという事は, 会話を行なっている者同士が, common ground を達成していることを示すための証拠となりえる. 文献[33]にその例がある.
- 明確な話題の開始(explicit topic opening)の使用回数: 明確な話題の開始は指示代名詞と反対に, メッセージの要点が, それを参照することを確立する句によって, 前に述べられていることをいう. 文献[33]にその例がある.
- breakdown: 参加者がコミュニケーションツールのことに言及するときに発生するものとして, 定義される. ここでのbreakdownは, コミュニケーションツールについてのことで, 単にメディアの限界を取り扱っているだけであるのに対して, 社会的なインタラクションにおける会話のルールの適用に伴うbreakdownについて言及していない.
- topic mention(話題の言及)の距離: 同じ話題の言及の距離は, それらの参照の間のターンの⁷個数として定義する. Voice ベースと比べて Text ベースの会話の場合, 距離が長くなることが予想できる. なぜなら, Voice ベースだと, その場だけの会話となってしまう, 過去の発話を再検討する方法がないからである.
- 発話の長さ, 発話の重複, 発話の中断: 普通の対面の会話は, 重複する会話と中断といった規則正しい出来事でターンが交替する.
- 凝視: アクションレコーダー⁸を用いると, 凝視を4つに分類することが可能である.
 1. 映像を見つめる.
 2. 紙の記録を見つめる.
 3. 計算機のスクリーンを見つめる.
 4. それ以外の場所を見つめる.

Sellen は文献[34]で, メディアの相違によるコミュニケーションの効果の違いを測定するために, パラメータがひとつだけ異なった実験同士を比較分析している. Sellen は, 会話構造を獲得するためにターンテイクに着目し, 以下の定義を行ない, 発話の切替を自動的に検出する方法を提案している.

ターン: 1 ターンは, 発言権を持つ人による沈黙と発話のシーケンスからなる.

話者交替: 話者交替は, 話者が発言権を失うか, 他の人が発言権を獲得したときに発生する.

同時会話: 発言権を持たない複数の人によって同時に発生する会話のこと.

⁷発話者の切り替わりのこと

⁸各参加者の発話の形跡を組み立てることを可能にする.

そしてその方法を用いて、対面式とビデオを利用した場合の相違を分析している。Sellen は実験前に、対面の方がビデオを介した場合よりもターンの回数が多く、ターンの平均持続時間が短く、ターンの分散度が小さいといった仮説をたてた。実験を行なった結果、その仮説に反して、ターンの回数やターンの平均持続時間、ターンの分散度には差異がないことがわかった。また、発言権を効果的に獲得するために、現在の話者の映像を他の参加者に切替えて表示させる方法を提案している。

映像技術の発展によって、作業者が物理的に同じ空間で作業する際に共有するものを、遠隔にいる作業者同士が模倣することが可能となる。Sellen は、研究者が理解する必要があるのは、映像がどの程度まで、その共有の提供が可能であるかといったことだとしている。また、映像を用いたインタラクションが会話の振舞いを変えようという可能性を含みながら、映像を用いたインタラクションは、同じ物理空間を共有することとは基本的に異なっていることも理解する必要があるとしている [36]。また、科学技術による遠隔の会話の効果を分析するために、以下の 2 つの実験を行なっている。

- 対面 vs 遠隔 (計算機を介した映像有り)
- 遠隔 (計算機を介した映像有り) vs 遠隔 (計算機を介した音声のみ)

この実験を分析する際に、コミュニケーションメディアが時間的な会話の構造にどのように影響を与えるかに注意を払っている。分析データは、参加者による発話パターンのオンオフを自動的にトラッキングすることによって獲得される。結果として、対面と遠隔の相違は、対面は相手の作業の邪魔をするという傾向にあるといった点にあり、映像と音声には差異がみられなかったとしている。しかし実験の参加者が、映像が会話の際には重要で利益があると感じたと示している。

文献 [37] では、共同執筆の作業タスクを 3 つの段階、planning phase, execution phase, integration phase に分類し、それぞれの段階におけるコミュニケーション形態の効果を分析した。以下にそのコミュニケーション形態を示す。

1. 対面
2. 計算機を介した環境 (E-mail + 1 対多の電子会議システム)
3. 計算機を介した環境 + 電話

planning phase では、計算機を介したコミュニケーションは、あまりうまくいかないことが明らかになった。さらに、計算機を介した環境に電話を導入しても、あまり効果がなく、電話で必要となる 1 対 1 の通信といった制限による困難さが付加されるにすぎないとされた。上述したように、実験から得られる結果がタスクに非常に依存しているといった観点から考察すると、タスクの種類によって効果の異なるコミュニケーション形態を分析することは、より厳密な結果を引き出すことが可能になると考えられる。ただし、そのフェーズへの分類するためのスキーマとそのフェーズの特徴について明確に認識しておくことは、重要な問題であると思われる。

3.3 作業や会話の調整

もし対面の会議であれば、参加者は身ぶりやアイコンタクトなどを使って、相手の行動や意図に気づくことができる。計算機を介した会議の場合、そのような mutual awareness や context awareness が簡単に獲得できるかは疑問である。作業の調整方法は、作業環

境で異なることが予想され、その環境が複雑になればなるほど、その複雑さを効率良く処理するメカニズムが必要となることが予想されている [38]。そういった観点から、CSCW の研究者の間では、発言権の制御が重要なトピックとなっている。対面の作業では、各々の作業者が、誰が発言しているのか、誰が発言したいのか、誰が会話の話題に興味がないのかといったことを認識することができる。それを遠隔の作業で支援するために、計算機のウィンドウに対面の作業と同じ状況を構築するといった試みが行なわれている。しかし対面と計算機のウィンドウは物理的に異なるので、対面の作業と同じ効果が得られないことが多い。そこで各作業者の発言権を制御し、会話を促進する方法が多く提案されてきている [39]。

文献 [40] では、個々の作業活動と集団の作業活動の間の関係の本質は何であるか、特に、作業環境における時間的、空間的なパターンの特徴が、どのようにして、協調的なふるまいに関連しているのかを分析をしている。その分析は、作業を終了させるための中断されない時間を持つことと、共に作業している人に話しかけるためのアクセスを持つことをの間の衝突を解決することから始められた。

3.4 作業効率と成果物の品質

共同作業の成果物を評価するための尺度として、以下のものが提案されている [38]。

- コスト: 時間, 費用, 精神負担
- 個々の報酬: 個人が獲得する報酬 (満足, 習得機会, 他人からの認知度とか) が、自分の作業の手間よりも多いと感じるか感じないか。つまり、個々の感情は、アクセス容易性と行動の制御性のもとで、分類することが可能である。
- グループ可能性: グループの結合力や、他の参加者と再び作業することを快く思っているか。結びつきのゆるいインタラクションであるのであれば、生産物の品質はあまり良くないといえる。

実際、作業効率や成果物の品質を評価することは非常に難しい。困難さには様々な原因が考えられるが、特に作業者間のスキル差に問題があると考えている。なぜなら、個人間のスキルの違いは必ず存在し、そのスキルの違いは、直接に作業効率や成果物の品質に影響を与えるものであるからである。このスキル差を小さくするために、多くの実験的研究では、実験タスクに関することを、授業等で共通に学習した学生を被験者とする傾向にある。また、実験タスクをプログラミングとした場合などは、プログラマー歴が同じ作業者を被験者にする傾向にある。

Chapanis らによると、すべてのコミュニケーションが記述されなければならない条件 (Text 型コミュニケーション) と、参加者がお互いに声で会話を行なうことが可能である条件を比較したときに、成果物に重要な相違があったとしている。しかし、声だけの場合と声 + 映像の場合を比較したところ、差異がみられなかったとしている [13]。このことは、人々是对面の方が電話よりも好むといった考えに反すると考えられるが、文献 [33] では、それに関して以下のように理由をあげている。

実際の作業では、優先順位のついた様々なタスクを持っていて、2 番目の順位のタスクのコストや、予想される労力から、1 番目の順位のタスクを保護する傾向にあるからである。つまり、タスクパフォーマンスに関するプロセスを測定するためには、1 番目

のタスクを保護するコストが必要となるのである。Chanpanis らの分析は、1 番目の順位タスクのみの分析を行っており、差異がみられなかったのは驚くことではないと考えられる。このことは、計算機を介したコミュニケーションの研究で報告された重要な効果が、成果物というよりむしろ、コミュニケーションプロセスを分析する尺度についてであったことを説明するのに十分であろう。つまり、タスクを被験者に与えて実験を行なう場合、作業品質や効率を計ることは困難であり、それを計る場合、現実の作業と同様な優先順位等をタスクに与える必要があると考えられる。

4 共同ソフトウェア開発

ソフトウェア開発とは、あいまいなユーザー要求を、不確定な要素を次第に減少させつつ、最終的には正確・高速ではあるが、一切曖昧性のないプログラムとデータに変換していく作業である。ものような作業は、初期の誤りが、後段に大きく影響するための危険度が高い。よって作業全体をいくつかの順次的フェーズに分解し、1つのフェーズにおける作業の完成度を十分に確認した上で進むことがよいといったことから生まれたのが、ウォーターフォールモデルである。しかし、このウォーターモデルの問題点が以下に指摘されている。

- 要求分析、獲得の段階で顧客要求を完全に理解することは不可能であり、プログラム設計の段階になって、認識されることがある。
- 仕様の決定時には、種々の対象分野に依存する方法論に従ってシステムの構造を作成するが、構築中のシステムの良否、不備な箇所を設計時に確認することは容易ではなく、コード作成後に検査するほかない。

以上の問題点を解決するために考案されたのが、

- プロトタイピングパラダイム⁹
- 操作型仕様パラダイム
- 形式仕様・自動変換パラダイム

などである。しかし、それらのモデルによって、ウォーターフォールモデルの問題点をすべて解決するわけではない。例えば、顧客と設計者、設計者同士などのインタラクションとして具現化される共同作業に起因する問題点などは、上記のモデルによっては解決されない場合もある。

そこで、ソフトウェア開発における共同作業に依存する側面を分析することで、その有効な支援方法が考察できるはずである。本節では、以下に示すような側面からソフトウェア開発を分析し、次節で、ソフトウェア開発におけるグループウェアの適用事例の考察を行なう。

- タスクの種類
- プロジェクトチームのメンバー構成
- 作業を行なう環境

4.1 タスクの種類

共同ソフトウェア開発の各フェーズについてまとめ、それぞれのフェーズにある問題点とその解決案を整理する。これによって、計算機によって支援を行なうための集団作業の基本的な性質を理解することができ、より効果的な支援について考察することが可能になる [2]。さらに、それぞれのフェーズの特徴を 2.1 節で述べたような McGrath のタスクの分類を参考にし考察する。

⁹CASE(Computer-Aided-Software-Engineering) が、プロトタイピング技法の重要性の認識とともに充実している。

4.1.1 Requirements Elicitation(要求獲得)

システムエンジニア (SE) が、顧客の要求および対象分野を分析して、様々な制約条件の中で具体的な仕様として明確にしていく過程を要求 (仕様) 獲得と呼ばれている。この要求仕様は、利用者と提供者の間のインタフェースと位置付けることができる [41]。

要求獲得 (requirements elicitation) フェーズでは、通常、顧客あるいはユーザーと開発者が数回の打合せを行なう。その打合せの参加者は4、5人といったところで、10人には満たない。そのため、並んで座っているもの同士が、全体で議論されているものと異なった特別な話題を議論していることはない。第一回の打合せで、顧客は初めの要求を説明する。打合せ後、開発者は要求に対する解を作成し、次回の打合せに顧客にそれを提示する。その後顧客と開発者は、同意に至るまで数回の打合せを行なう。以上のように要求獲得フェーズは以下に示すようなサブフェーズの繰返しから構成されている。

- 顧客が要求を説明するフェーズ
- 開発者が顧客に解答を提供するフェーズ
- 顧客と開発者が同意をとるフェーズ
- 開発者のスケジュールの調整をするフェーズ

4.1.2 Specification Development(仕様の決定)

仕様化設計技法は数多く存在するが、それらのほとんどが、グループのためではなく、個々の作業員のためのものである。そこで、グループに適した支援ツールや方法を構築する必要がある。グループによる仕様設計の場合、各々の作業員は他人と異なった方法によって仕様の部分を設計する可能性がある。例えば、ある作業員がデータフロー図を書いていて、相手の作業員が状態遷移図を書いてい場合がある。この場合、どのようにして、異なった作業員が設計した仕様の部分を一つに統合し、統合の際に発生する矛盾をチェックするのかが、解決されるべき問題点である。

文献 [42] では、様々な作業員によって決定された異なる仕様の統合の問題に関して考察している。具体的には、部分的な仕様を保持し、異なったスキーマや開発方法を利用しながら、記述、開発される多様な ViewPoints を開発している。この ViewPoint は、5つのスロットを持っている。

- 記法
- 作業方法
- ドメイン (対象)
- 仕様結果
- 作業履歴

そして、異なった ViewPoint の間の関係を表現するために、内部 ViewPoint ルールの利用を探求している。

4.1.3 仕様のレビュー

文献 [43] から、仕様のレビューについて以下のようにまとめる。生産物の品質の向上は、作業を行なう際の目標であるが、人間的要因のかかわりの多いソフトウェア開発で

は、深刻な問題となる可能性がある。開発の上流工程にあたる要求分析や要求獲得を基本設計から、下流工程の製造・テストにいたるまで、すべての面で品質が保証されなければならない。コーディングのわずかなバグが重大な問題を引き起こすこともあるが、要求仕様や設計の誤りは開発したシステムを全く使いものにしなくする可能性がある。このような誤りを防ぐ手段として、レビューというものが行なわれている。このレビュー作業は、ソフトウェア工学の成果が実際の開発に適用され、ソフトウェア開発の方法が発展しようとも、要求仕様や設計に対するレビューの重要性が変化することはない。またこのレビュー作業は作業員間のコミュニケーションが発生しやすく、人間的要因が大きく影響すると考えられる。またレビュー作業には、外部の人を設計チームに加えてレビューを行なう公式レビューと、外部の人を加えず、設計チーム内だけの人でいつでもレビューを行なう非公式レビューがある。

4.1.4 Coding(プログラミング)

大規模なソフトウェア開発を行なう場合、行なうタスクは複数に分割され、各々の作業員が割り当てられたプログラムを構築する。そのような共同プログラミングではタスクは個人作業に分担される。しかし個人作業にも限界があり、それぞれのプログラムの摺り合わせ作業を行なう必要がある。そのような場合、ツールとして共有エディタが使われやすい。ツールとして共有エディタが使われやすい。他のタスクとは異なって、作業員は自分に分割されたタスクに責任がある。単に自分のソフトウェアモジュールを開発し、修正すればよいとされている。よって、データへのアクセスの同時性制御は、責任の管理や、責任のない作業員にはモジュールをみせないといったことから実現できる。

文献[44]では、コーディング段階で、スキル差のあるプログラマに、プログラミングをしてもらい、スキル差から発生する影響を分析し、エキスパートのプログラマが以下の2つのタイプのプログラミング知識を持っていることを確かめている。

programming plans: プログラミングにおいて類似する動作を表現する一般的なプログラムの断片。

rule of programming discourse: プログラミングのルールを理解し、プログラム断片をプログラム全体に構成する決まり。

文献[45]では、プログラマが新しい計算機言語やプログラミング環境、ソフトウェアフレームワーク、システムに直面した時に発生するであろう、熟練者に聞くといったインフォーマルなインタラクションについて分析している。著者らは、熟練者によって提供される様々な困難-発見のための情報、付随して習得する知識、会話を管理するための方法、個々の生産性について記述している。それらの結果から、どのようにすれば、計算機支援ツールによって、初心者も、タスクの状況を要約しサブタスクに移行、協調インタラクションを促進、そしてコンサルタントを補うことが可能になるのかを議論している。

4.1.5 Code Inspection

Code inspection(コードインスペクション)とは、注意をいくつかの選ばれた局面に払い、一度に1項目に制限して、迅速に、コードを評価する作業のことである。幾人かの人々によって行なわれたコードの細部を、コード全体のものとして評価することである。

4.1.6 Debug and Test

コーディングとテストの作業は、労働集約の典型となる。個々の作業者は、デバッグ、テストされるモジュールに責任をもち、統合したモジュールテストを行なう場合に、それぞれの作業者と、コミュニケーションする必要がある。デバッガーやテストのためのツールに、コミュニケーションツールを統合するべきである。

またデバッグやテストを行なう際に、設計問題解決に至る過程や設計議論の内容などの設計履歴情報は有効かもしれない。なぜなら、そのような情報はデバックやテストをしたときに発見されるプログラムのバグなどの発生原因を示すものとなりえるからである。それに関する支援に関しては5節で詳しく説明することにする。

4.1.7 Maintenance

共有されているソフトウェアモジュールを修正する場合、その共有の度合いが高ければ高いほど、慎重におこなわなければならない。たとえば、不用意な修正は、それを利用している一部のシステムに矛盾を引き起こすことになる。ソフトウェアの適用する業務は日々自律的に変化するのに対して、また人工物であるソフトウェア自身は基本的に手作業で修正/変更して行かなければならない。

4.1.8 各フェーズのタスクタイプ

上述した各フェーズのタスクを、2.1節で説明したタスクの分類を参考にして分類を行なうと、表1ようになる。この分類からもわかるように、各々のフェーズは、1種類以上のタスクタイプが混在している。これは、各フェーズのタスクの複雑であることを反映している。

表 1: 各作業フェーズのタスクタイプ

	要求獲得	仕様の決定	仕様のレビュー	プログラミング
Planning tasks	○			
Creative tasks		○		○
Intellective tasks	○		○	○
Decision-making tasks		○	○	
Cognitive tasks	○	○		

4.2 グループのメンバー

近年の科学技術の発展とともに、ソフトウェアも大規模なものになってきている。その大規模ソフトウェアを開発するためには、それに従事するメンバーを多くする必要がある。そのような開発チームを考えた場合、そのドメインに熟練したものもいれば、その設計対象について熟練したものもいる。また、プログラミング言語やソフトウェア開発環境についてスキルのある作業者がいることもある。そういった多様なスキル差をもった作業メンバーによる、共同作業は、同じスキルをもった作業メンバーによる共同作業

と同じということは考えにくく、そのスキル差についても、これから考えていくべきである。

ドメインエキスパートの知識、技術の役割がどうなっているのかを分析した文献 [46] がある。ソフトウェア設計者の基礎をなす知識や技術、それらがドメインの経験に影響を与えるための方法を分析するために、様々な設計状況を実験的に設計した。また初心者と熟練者との振舞いの相違を分析することにより、熟練者の知識や技術の利用方法について考察している。

4.3 作業環境

共同ソフトウェア開発において通常行なわれる作業環境は、4.1節でまとめられている各作業フェーズに依存することが予想できる。例えば、要求獲得フェーズでは対面の作業で行なわれ、下流工程であるコード設計に関するフェーズでは、個人の作業が増加するので、作業同士は非同期の作業を行なわれるといったことが考えられる。

しかし、産業構造や業務範囲の拡大により、それぞれのフェーズで最適の環境を選択できない場合もある。例えば、顧客と設計者が別の地域に居住するために、要求獲得フェーズにおいて、遠隔リアルタイム電子会議を利用しなければならない場合も出てくる。また、視点を変えて考えると、従来それぞれのフェーズで要求された作業環境や作業様式、特に他者との関わりによって決定されているものなどは、グループウェアなどの導入により自由に変更することが可能だと思われる。

共同ソフトウェア開発において、計算機で支援できると考えられる具体的な作業環境を以下に示す。これは2.3節にあるようなJohansenによるグループウェアの分類と同じである。

- 同じ時間、同じ場所 (対面同期)
- 同じ時間、違う場所 (分散同期)
- 違う時間、違う場所 (分散非同期)
- 違う時間、同じ場所 (工場等のシフト作業)

5 共同ソフトウェア開発支援

共同ソフトウェア開発では、グループ内コミュニケーションを円滑にすることが重要である。よって、コミュニケーションを支援するグループウェアはソフトウェア開発支援に、重要な役割をもつ。文献[47]では、1992年頃までに行なわれた、グループウェアのソフトウェア開発への適用の事例や研究が紹介されている。

本節では、共同ソフトウェア開発へのグループウェアの適用に関する研究や事例を以下の観点から整理する。

- 従来の作業の分析事例
- 支援方法, 支援ツール
- 共同ソフトウェア開発のためのグループウェアの評価

5.1 共同ソフトウェア開発のケーススタディ

文献[10]では、ソフトウェア開発のケーススタディを行なっている。3.1節にも上述されているように、共同ソフトウェア開発の初期フェーズに焦点をあて、グループの活動の分析を行なうことにより、効果的な支援方法を探求している。また文献[48]において、Curtisは、MCCで行なわれた設計プロセスに関する研究を概観し、これから必要となる研究を探求している。ひとつの結論として、プロジェクトの結果は、プロジェクトを行なう前から決定されているとしている。

5.2 支援方法, 支援ツール

Curtisはソフトウェア開発へのツールの適用を特定すれば、より効果的な支援が可能になると指摘している[48]。佐伯はソフトウェア開発において集団作業の特徴は各作業フェーズで異なり、実験的研究を通してその特徴を分類し、各々のフェーズに適したツールを開発すべきだとしている[49]。つまり、ひとつのツールを用いて、ソフトウェア開発のすべてのフェーズを支援することは不可能だとし、ツールを統合し、適切なツールを選択できるような環境が必要だと述べている¹⁰。佐伯はまた、文献[51]において、実験的研究からソフトウェア設計フェーズにおける電子メールを使用する利点、欠点について分析している。著者らは、ソフトウェア開発の仕様作成の段階の支援するためのツールを提案している。特に仕様作成を以下の3つのフェーズにわけ、

1. チームの作業プランを作成する。
2. メンバーがそれぞれ個々に、割り当てられたタスクを行なう。
3. メンバーによって開発された生産物をレビュー、統合する。

2のフェーズの支援を対象にしている。佐伯らは、コミュニケーションは共同作業の重要な特徴の一つであり、その特徴は、共同作業者が従事している作業フェーズ毎に異なるとした。そして、その特徴を、実際のコミュニケーションの観察から抽出し、フェイズ毎の支援方法について研究を進めた。その結果をもとに以下に示すような2つの部分からなる支援ツールを構築した。

¹⁰Ellisは、それを達成するための概念のひとつとして、グループウェアを概念的な3つのモデル(ontological model, coordination model, interface model)を用いて分類し、この3つのモデルによる記述に適さないグループウェアをメタグループウェアと呼んでいる。簡単に言えば、このメタグループウェアはクラスであり、グループウェアはインスタンスである。[50]。

1. Method base system: 作業者は、様々に分類された一覧から、設計ドメインに適切な設計法を選択し、個人作業を行なう。
2. Structured E-mail system: 生産物にメッセージやコメントを添付してそれを電子メールとし送・受信することにより、共通の部分に関与する設計者同士のコミュニケーションを支援する。ここでの電子メールは、いくつかのテンプレートを持つ構造化メールを利用することができる。このメールは、生産物の変更などに応じて自動的に送信される。

佐伯らはまた、発話行為(スピーチアクト)理論 [18] を用いて、要求獲得の記録方法について分析している [52]。参加者の発話の特徴は、要求獲得中でさえ、フェーズによって異なるため、それぞれのフェーズ毎に記録する構造を定義している。例えば、顧客が開発者に要求を説明するフェーズでは、顧客が自分の要求を開発者に提示し、開発者はクリアではない箇所を明らかにするために質問をするという特徴がある [53]。佐伯らは、そのようなフェーズは質問-解答といった構造で会話を記録すること提案している。各々の参加者は、顧客、開発者といったような役割をもっており、発話が与える振舞いは、役割に依存しており、彼らは自分達の内部状態を持っているが、その状態は、相手からの発話を受け取ることによって変化する。発話行為は、同じ状態の遷移のパターンを引き起こす発話を抽象化したものとみなすことができる。例えば、参加者が発話行為が「提案」である発話を与えたとすると、聞き手の内部状態は、その提案を理解すべきであり、自分の立場を明白に表現する状態に変化する。この発話行為と状態遷移は、参加者が参加しているフェーズに依存している。発話から発話行為を認識することは、話者の意図を引き出すことに有効である。この発話行為を発話から取り出す方法として、

- キーワードやキーフレーズの利用
- 典型的なパターン(質問-解答は隣接ペア [16] など)の利用

を提案している。

文献 [53] では、シナリオに基づいた要求分析を行なっている。この論文では、Inquiry Cycle Model を提案しており、それは、発話行為理論を使った佐伯らの方法 [52] と類似している。それは、要求獲得を質問-解答サイクルとみなし、質問の分類に焦点をあてている。

文献 [54] では、顧客と開発者との間の情報交換の方法を分類するために、実験的研究を行なっている。その方法のことを著者である Keil らは、“links” と呼んでいる。この links は、顧客と開発者が情報交換を行なうことを可能にしてくれる技術やチャネルとして定義されている。そして、“custom” と “package” 開発環境とを比較することによって、適切な links について議論している。ただし、custom は顧客が開発者に発注するようなソフト、package は世間に売り出すようなソフトのことを示している。それらの結果は実用的であり、現在ある links について議論、評価するのに役立つ。つまり、それらの links をどう利用するかを決定するための理由となるとしている。

ソフトウェア開発の仕様を決定作業を 4 つの構造に分類している文献 [55] がある。分類した 4 つの構造は、negotiation, integration, delegation, replication であり、その構造の変移パターンについても分析している。

文献 [56] では、要求獲得のための支援ツールを提案している。このツールの支援対象は対面作業であり、対面の行なわれる顧客と SE との間で行なわれる要求分析会議で。

顧客と SE との間で発生する認識の違いを解消することを支援する目的で開発された。

文献 [57] では、利用者要求文書 (User Requirements Documents:URD) における欠陥の発見を、N-fold 点検方法を用いることにより向上させるための実験的研究を行なっている。ソフトウェア開発プロセスの初期段階では、URD の徹底的な検査を行なう。URD には、提案されるソフトウェア生産物の機能性と性能の特徴が、利用者によって記述されており、利用者と開発者の同意をもって成立する。完全で正確な URD はあらゆる設計に不可欠であり、初期段階に発見されなかった省略や、不調和、あいまいさ、矛盾は、ソフトウェアの設計段階が進むにつれて大きな問題となり、修正するのに莫大なコストがかかってしまう。そこで文献 [57] では、完全な URD を作成するための N-fold 手法について検討している。N-fold 手法とは、集団のメンバーを調整し責任をもつ一人の調整者と N 個の独立した集団が存在し、N 個の別々の集団が点検を並行に行なうといった手法である。この手法の評価実験から、ある程度までは欠陥が発見できたが、発見できないところがあり、その箇所については、設計、実装、テスト段階で発見されなければならないだろうとしている。

ソフトウェア履歴利用の研究動向に関しては、文献 [58] がよくまとめられている。ソフトウェア開発の効率化、よりよい設計判断の導出や設計空間の再利用などを目的として、設計問題解決に至る過程や設計議論の内容などの設計履歴情報のある構造的な枠組みを用いて、獲得、整理、表現、保存する技術を Design Rationale¹¹(DR) 技術という。以下に DR 技術をささえる代表的なモデルをあげる。

- IBIS モデル (gIBIS etc)
- QOC モデル
- SYBYL/DRL モデル

gIBIS は、Conklin らによって、実際の柔軟な利用状況を考慮して IBIS モデルにいくつかの拡張をおこなったものである [59]。これには、以下に示すような問題点がある。

- 決定基準 (Criteria) 空間を明示的に表現できない。
- 決定問題の依存関係および階層的な関係が表現できない。
- 決定事項とそれから生成される成果物 (DFD 図とかソースコードなど) と議論過程の関係が表現できない。
- 直接議題とならないような議論や質問を表現できない。

この問題を解決するために、Potts と Bruns モデルや、PHI モデルがある。以上が IBIS 系のモデルであり、議論指向プロセス指向のモデルといえる。

QOC(Question, Option and Criteria) は、以下のノードから構成されており、空間構造化指向であるシンプルなモデルである。

- Question ノード: 設計空間のキーとなる課題を示す。
- Option ノード: 解決案の候補を示す。
- Criterion ノード: 比較評価の基準を示す

SYBYL は、QOC の議論空間構造化指向と gIBIS の議論プロセス指向モデル指向のモデルの両方を取り込んだものである [60]。SYBYL はグループ意志決定支援を目的に開発されたシステムであり、知識の共有と意志決定支援機能をもつ。それは、議論空間や

¹¹設計理由とかソフトウェア履歴情報と呼ばれている。

議論プロセスの表現性に優れた記述言語 DRL をもつ。DRL における代表的なノードは以下のとおりである。

- Alternatives: 問題解決の候補
- Goals: 解決策が満足すべき性質, 目標 (要求条件)
- Claims: 個々の解決策の選定にあたっての背景議論

あと補助的なノードとしては, 以下のとおりである。

- Question: 単なる質問を表現
- Procedure: 質問を答える時の手順

文献 [61] では, ソフトウェア開発で重要な部分というのは, 成果物に至る過程であり, それらに対する焦点がソフトウェア工学には欠けているということを示し, DR による解決策を提案している。ソフトウェア開発において, ウォーターモデルといったようなソフトウェアプロセスモデルは, 多くの目的を持っている。しかしトップダウン型の実行は, 開発を困難にさせることが示されていて, この著者らは, マネージメント責任モデルの特徴というよりは, 理解しやすいようなレベルで, ソフトウェア開発を表現するための様々なプロセスモデルのタイプを提案している。

DR とは異なるが, ドメイン指向型設計環境における意図やコンテキストを明示することによって, 長期にわたる非同期協調作業が支援されるといった例を示した文献 [62] がある。意図は, デザインタスクの基となる目的, ゴールである。その中には, 満たすべき制約や資源など, 色々な情報が含まれている。コンテキストとは, そのような意図やデザインアーティファクトそのものを表出する際に用いられる背景となっている知識や情報である。そこでは, 意図やコンテキストを明示化することによって,

- 人間と計算機のインタラクション
- 個人利用者による設計時の意図の認識
- 非同期協調作業における人間同士のコミュニケーション

に有用であるとしている。

DR とは違う概念であるが, DR と同様に情報を分類し, グループ内の共有を支援するものとして文献 [63] がある。著者らは, 目的に応じた semi-structured なメッセージのテンプレートを活用することにより, メッセージの優先度付けや分類を自動的に行なえるようにしたシステムを開発した。このシステムは, Information Lens と呼ばれている。これによって, 電子メール, NetNews における情報のオーバーロードを克服し, グループにおける情報の共有を促進することが可能になるとしている。

発想支援ツールを用いた要求仕様分析を行なった研究がある [64]。要求仕様のあいまいさを収束させる作業において, 特定の方法論に依存した CASE ツールの利用は効果的ではないといったでの指摘 [65] から, カード操作の利用を提案している。

文献 [66] では, ソースコードの検査を支援する対面同期型のグループウェアを提案している。このツールは CASE ツールとしても興味深い, 以下のような特徴もある。

- コード検査会議における司会者, 検査者, 書記などの役割を支援する。
- ペーパーレスの検査作業を行なう。ソースリストのハードコピーも紙への記録も行なわない。
- 検査者は自分のマシンで検査作業を行なうことができる。

5.3 共同ソフトウェア開発ツールの評価

文献 [67] では、共同ソフトウェア開発のスケジューリングをタスクの対象とし、ウォールボードを用いた場合と計算機で支援されたスケジューリングツールを用いた場合との比較分析を行なっている。結果として、ウォールボードを用いた場合の方が利用性が高いことがわかった。この原因として、

1. ボードの大きさが視覚的に有利である。
2. ボードは公共性があり、スケジューリングの責任が発生し、コミュニケーションが活発になりやすい。
3. ボードは物質であり、人間が物理的に操作することによって、スケジューリングの活動をよく理解することが可能である。

といったことを述べている。また、スケジューリングの詳細な活動フェーズとして、

1. 初期の計画
2. スケジュールの更新
3. スケジュールの再計画
4. 集団作業の調整

をとりあげ、それらのフェーズにおけるウォールボードの利点を説明している。そして、この利点をふまえたグループウェアの設計をすべきだと筆者らは主張している。この論文において、共同作業のタスクをソフトウェア開発のスケジューリングに焦点をあてているのは興味深い。

これによって、比較評価の基準 (Criteria) を明示的に示すことが可能である。QOC 言語を用いて、DR 文書の経験的評価を行なった文献が [68] である。設計対象の DR は、フランス航空宇宙会社の機械設計部門における再設計作業の初期段階の 9カ月の作業から獲得している。DR 文書の評価は、以下の 3つの点から行なっている。

1. 設計者が理解出来ない設計に直面した場合、DR を知ることが必要かどうか。
2. 設計者は DR 文書を使用するかどうか。
3. 設計者の必要とする理由を獲得できるかどうか。

分析結果であるが、DR 文書は少なくともそれを多く読む作業者にとっては、以下に示すように、役に立つであろうが、十分ではないといったことが得られている。

1. 設計者の質問の半分以上が DR に関することであった。
2. DR 文書を広く利用して、以前の設計の理解や評価の助けとする設計者がいた。
3. QOC 文書では、設計者に関する DR 疑問の半分以下しか答えられなかった。

以上より、DR 文書の限界が明らかになったとしている。

DR をソフトウェア開発に利用した研究としては、文献 [69] がある。この文献では、箇条書きのテキストを用いた単純な DR 記法である itIBIS を利用して、18ヶ月以上に渡るソフトウェア設計の議論の経過を記録している。この著者らは、あるアクション (活動) が何故発生するのかに関する情報が重要であるにも関わらず、大規模プロジェクトでは失われてしまっていることを取り上げ、その理由を、それらの情報があまりにも構造的ではなく、即座にそれを獲得、検索することが困難であるからだとしている。そこで簡単な構造化方法、IBIS を上述したような情報を記録、検索のために利用することに

よって、ハイパーテキストやグループウェアの意味や、ある技術を用いた設計プロセスを支援するための見込みについて議論している。具体的には、issueに基づいた技術が、以下の3つのことを処理することを導いている。

- グループ決定支援: 解を決定するのが困難ではない。新たな問題点を設定し、新しい解を決定することが可能となる。IBISは解の質を向上させる。
- 会話の構造化: 会話はグループのパフォーマンスにとって、とても重要な側面をもっているが、それを構造化するといった作業は、作業者にとって邪魔なものであり、好ましいものだといえない。しかしIBISが、事後の情報を構造化するために利用されていたといったことは、注意すべき点である。
- グループ中の記憶管理: 設計理由を構築するIBISは、ひと続き(時間軸上)の情報を、系統的に整理され、リンクが交差していて、それぞれが関連しているような構造をもっていて、グループの記憶装置となる。

計算機はこれらのDRの抽出、表現、記録の手間を代行してくれる可能性があるが、言語構造が複雑すぎると使う側に負担を強いることになる。このような問題についても考察すべきであり、QOCのような単純なDRが良いといった考え方もある。

共同ソフトウェア開発のためのグループウェアとしては、ソフトウェアの設計履歴(DR)を利用した非同期型グループウェアが多く開発されてきており、成果をあげているといえるだろう。しかしながら、共同ソフトウェア開発における各々のフェーズに焦点をあて、作業者間で交わされるコミュニケーションについて分析している研究は少ない。共同ソフトウェア開発においてグループウェアが効果的に利用されるためには、ソフトウェア開発といったタスクに依存した効果的なコミュニケーション方法を導く必要があると考える。

その方法として、実世界の作業フェーズについて分析し、それぞれに適したコミュニケーション支援を開発/選択するのが有効であると考えられる。しかし実世界の作業は、現存のコミュニケーションツールによって制約を受けた環境であると解釈できる。既存の方法とは異なるコミュニケーション手段を提供することによる作業の促進についても同時に考慮すべきである。

6 まとめ

6.1 実験方法/評価手段の考察

CSCW の研究者の間では、大学や研究所の実験室で行なわれている実験的集団作業について論争されている。たとえば、組織が作られた背景や作業ノルマなどは、研究室の集団作業では現実的であるとはいえない。しかし、研究室における集団作業の実験にも利点がある。たとえば、同じタスクを複数の集団で行なうことができ、生産物の品質だけでなく、作業プロセスを研究できる。この種の評価は、現実の世界の集団作業では不可能である。研究室の作業を現実の世界の作業に近づけるため、ある制限された時間で作業を行なう必要のある実際の問題を与えることが考えられる。それを実現している研究として文献 [10][11] がある。

また、実験的研究を進めることは困難であるといわれている。例えば、個人作業よりもグループや組織の作業が複雑であることや、集団作業を研究室で運営することはコストがかかってしまうといったことが困難の要因とされている [70]。また分析結果を得るためには、膨大な時間と労力がかかってしまう。以上のような困難さを解消するためには、過去の評価方法や評価結果の参照やグループ作業や技術の基礎的な特徴を理解することにより、他の研究との比較を行なうと良いであろう。

我々の研究目的は、効果的な共同ソフトウェア開発に重要なコミュニケーションに焦点をあて、それを支援するシステムを選択する手段を明らかにすることである。現在は、その目的のためにコミュニケーション支援システムの定量的な評価手段を検討している。具体的には、様々な作業環境や作業フェーズ、作業グループといったパラメータを考慮したうえで実験作業を設定し、コミュニケーションの分析に焦点をあてプロトコル¹²解析 [17] を行なっている。この解析法は、問題解決活動にかかわる個人や集団の活動を、テープレコーダやビデオで記録し解析することにより、人間の行動特性に関する種々の規則性を発見しようとする実験心理学の一手法である。実験では、タスクと作業グループは変化させずに、作業環境を構成するパラメータの値が一つだけ異なる環境で実験を行なった。研究室の実験を現実の作業に近づけるため、タスク遂行時間にある程度の制限を設けた。その一つだけ異なる環境同士を比較することによって、そのパラメータによるコミュニケーションの影響を分析し、そこで発生するコミュニケーションの特徴を明らかにした [71][72] [73][74]。

文献 [71][72] では、ネットワークを介した共同作業における阻害について、文献 [1] で言われている、計算機で支援される環境を構成するような3つのパラメータの作業への影響を分析するために、パラメータの値が1つずつ異なった実験環境を設計し、実験を行なった。この方法は、Sellen がメディアの相違によるコミュニケーションの影響を分析するために行なった実験でも採用されている [36]。コミュニケーションを定量的に評価するために、会話の長さなどといったコミュニケーション変数を導入した [35]。さらに会話のターンテークに着目し、コミュニケーションの分析を行なった。その結果、非同期型コミュニケーションでは、会話の長さ、話題の複雑さといった変数が同期型の場合と異なり、話題の完結性に関係があること予想できた。そこで、コミュニケーションの同期性に着目し、同期/非同期といったパラメータを用いた実験を行ない比較分析

¹²人が自分自身の知的営みについて語ること(語らせられたこと)であり、その記録のこと。

を行なったのが文献 [73][74] である。

文献 [73] では、同期型と比較した非同期型コミュニケーションの特徴について言及し、その特徴が及ぼす作業の阻害について明らかにした。具体的には、会話、発話といったコミュニケーション変数を定義し、話題の完結性という尺度からコミュニケーションを定量的に評価した。特に非同期型のコミュニケーションの場合、この話題の完結性が、話題が並行に進められる度合（話題の複雑さ）によって影響されるといった特徴を導いた。また発話のやりとりの頻度によって、話題の完結性が変化することも導き、発話の大きさ¹³について考察した。文献 [33] では、コミュニケーションの common ground を測定する方法として、syntactic entrainment を測定することを述べていて、発話の長さ（発話の大きさ）というのが syntactic entrainment の尺度の一つとしている。例えばもし、長い（大きい）発話を利用してコミュニケーションを行なっている場合、相手も長い（大きい）発話を利用する傾向にあるといったことを示している。このことを参考にすると、非同期の場合に話題が完結しない原因として、コミュニケーションの common ground を達成できないコミュニケーション方法、つまり長い発話のやりとりを行なっているからだと考えることができる。

作業の阻害は、会話の完全性の観点から考察し、非同期型コミュニケーションとの特徴との関係について論じている。文献 [74] では、その会話の完全性と生産物の品質との関係を分析し、ソフトウェア開発の要求仕様分析フェーズにおいて発生する、要求仕様のあいまいさや仕様の変更に関して、会話の不完全が悪影響を及ぼすことを導いた。古くからの社会心理学の研究において、対面のコミュニケーションと計算機を介したコミュニケーションの比較を行なったものが数多くあるが [13][14]、Chapanis らは、実験から得られる結果は、実験タスクに非常に依存しているといったことを明らかにしている。またタスク遂行の効率の尺度を、どのくらい良く、どのくらい早く実験タスクを完成させるかといったこととするならば、それはコミュニケーションの利用のしやすさの変化の総計であるといったことが明らかになっている。そういった観点から考えると、我らが行なった実験の結果は、ソフトウェア開発の仕様のレビューといった作業に固有の結果であり、単純に、実験タスクの成果物だけを評価することは意味がないことがわかる。つまり文献 [74] で行なわれているように、タスクの成果物と、それに関わるコミュニケーションとの関係を同期/非同期で比較することは意味があると考えられる。

McCarthy らは、コミュニケーションの common ground やコヒーレンスの達成といった観点から同期型 Text のコミュニケーションの分析を行なっている [29][32][33]。common ground は、コミュニケーションを行なう作業者同士の共通認識のことを示しており、会話のコヒーレンスの達成とは、作業者同士がメッセージとメッセージのコンテキストを結び付けることができることを示す。話題の完結性を common ground の達成とコヒーレンスの達成を示す特性としてみなすことができるかもしれない。

今後の実験的研究を進めるにあたり、以下の 2 点に注意したいと考えている。

グループの規模：対象とする作業の人数と作業環境が依存関係にある可能性をもっており、作業人数によって、作業環境と作業への影響の関係が変化することを予想される。

対象タスク：現在の共同ソフトウェア開発における各フェーズの中で、コミュニケーシ

¹³ここでは、一回の発話で相手に伝えるメッセージの大きさのこと。

ンが発生しやすいタスクを明確にする。しかし、グループウェアの利用によって、今まで行なわれなかったコミュニケーションの可能性についても考えることができるので、コミュニケーションが発生しやすいタスクに限定する必要もないかもしれない。ただ、対象タスクは明確にする必要があると考えている。また、グループの規模同様、対象タスクによって、作業環境と作業への影響の関係が変化することが予想される。

6.2 自在システムにおけるコミュニケーション支援の評価

我々が行なってきたコミュニケーションの同期性の評価手段から、落水研が進めているソフトウェア分散開発環境「自在」[75][76]の「栞」[77][78]で発生するコミュニケーションについて考察を進めることも考えている。「栞」は蓄積型グループウェアであり、ソフトウェアプロセス実行における系統的コミュニケーション支援といった立場から、変換プロセス記述の詳細レベルに合わせたコミュニケーション記録の単位を定義し、全体状況の把握から設計根拠の理解までの目的に応じた情報を提供する。コミュニケーションの記録は、討議空間、討議プロセス、討議の型の3層スキーマで行なわれる。しかし、この3層スキーマによってコミュニケーションが記録されたとして、全体状況の把握や設計根拠の理解が可能になるかどうかに関する評価を行なう必要がある。また、それらが可能になったとして、実際にコミュニケーションの快適性を評価する必要がある。たとえば、我々が分析した非同期型コミュニケーションの特徴をそのスキーマに、考慮する必要があると考えられる。また3層スキーマの有効性を確認するために、決定事項の管理といった観点から評価することを考えている。例えば作業者の会話記録から、以下の分析を考えている。

- 決定事項の参照回数: 参照回数が多いと、作業者があまり認識していない可能性がある。
- 決定事項に関する議論数: 議論数が多いと作業者間で共通の認識がとれてない可能性がある。
- 決定事項への参照距離: 距離が長いと作業者間の認識が一致していない可能性がある。

そのようにして作業者間のコミュニケーションを分析することにより、決定事項が共通に認識されているかどうかを明らかにできると予想する。文献[68]で記述されているようなDRの評価方法も役に立つかもしれない。また、共同作業後に作業者に決定事項に関するアンケートを行ない、そのアンケート結果の相違を測定することによって、決定事項に関する認識の不一致度を測定することを考えている¹⁴。

¹⁴3.2節で述べた、common groundを測定するための方法を参考にしている。

参考文献

- [1] C. A. Ellis, S. J. Gibbs, and G. L. Rein. GROUPWARE some issues and experiences. *Commun. ACM*, Vol. 34, No. 1, pp. 39–58, Jan. 1991.
- [2] Judith S. Olson, Stuart K. Card, thomas k. Kandauer, Gray M. Olson, Thomas Malone, and John Leggett. Computer-supported co-operative work: research issues for the 90s. *Behaviour & Information Technology*, Vol. 12, No. 2, pp. 115–129, 1993.
- [3] 落水浩一郎. ソフトウェア開発におけるグループウェア開発の役割. ソフトウェアツールシンポジウム'92, pp. 83–92, 1992.
- [4] J. E. McGrath and A. B. Hollingshead. *Groups Interacting With Technology*. SAGE, 1994.
- [5] Joseph E. McGrath. *Groups: INTERACTION AND PERFORMANCE*. Prentice-Hall, 1984.
- [6] Robert Johansen. *Computer Support for Business Teams*. New York, 1988.
- [7] 石井裕. CSCW とグループウェア. オーム社, 1994.
- [8] Eevi E. Beck and Victoria M. E. Bellotti. Informed Opportunism as Strategy: Supporting Coordination in Distributed Collaborative Writing. In *ECSCW'93 proceedings*, pp. 233–248, Sep. 1993.
- [9] Christine M. Neuwirth et al. Computer Support for Distributed Collaborative Writing: Defining Parameters of Interaction. In *CSCW'94 proceedings*, pp. 145–152, Oct. 1994.
- [10] Gray M. Olson, Judith S. Olson, Mark R. Carter, and Marianne Storostern. Small group design meetings: An analysis of collaboration. *Journal Of Human-Computer Interaction*, Vol. 7, pp. 347–374, 1992.
- [11] Judith S. Olson, Gray M. Olson, Marianne Storosten, and Mark Carter. How a Group-Editor Changes the Character of a Design Meetings as well as its Outcome. In *CSCW'92 proceedings*, pp. 91–98, Nov. 1992.
- [12] Gary M. Olson, Lola J. McGuffin, Eiji Kuwana, and Judith S. Olson. Designing software for a group's needs: Functional analysis of synchronous groupware. In L. Bass and P. Dewan, editors, *User Interface Software*, chapter 7, pp. 129–148. John Wiley & Sons Ltd., 1993.
- [13] Alphonse Chapanis. Interactive human communication. *Scientific American*, Vol. 232, No. 3, pp. 36–42, 1975.
- [14] Ederyn Williams. Experimental comparisons of fact-to-face and mediated communication: A review. *Psychological Bulletin*, Vol. 84, No. 5, pp. 963–976, 1977.
- [15] Stephen Gale. Homan aspects of interactive multimedia communication. *Interacting with Computers*, Vol. 2, No. 2, pp. 175–189, 1990.
- [16] Charles Goodwin and John Heritage. Conversation analysis. *Annual Reviews Anthropol*, Vol. 19, pp. 283–307, 1990.

- [17] 海保博之, 原田悦子. プロトコル分析入門. 新曜社, 1993.
- [18] J.R. Searle. *Speech Acts*. Cambridge University Press, 1969.
- [19] R. L. Daft and R. H. Lengel. Organizational information requirements, media richness and structural design. *Management Science*, Vol. 32, No. 5, pp. 554–571, 1986.
- [20] M. L. Markus et al. Why CSCW Application Fail: Problems in the Adoption of Interdependent Work Tools. In *CSCW 90 Proceedings*, pp. 371–380, 1990.
- [21] Barbara L. Chalfonte, Robert S. Fish, and Robert E. Kraut. Expressive richness: A comparison of speech and text as media for revision. In *CHI'91 proceedings*, pp. 21–26, 1991.
- [22] Robert Kraut, Jolene Galegher, Robert Fish, and Barbara Chalfonte. Task requirements and media choice in cooperative writing. *Journal Of Human-Computer Interaction*, Vol. 7, pp. 375–407, 1992.
- [23] Robert E. Kraut, Colleen Cool, Ronald E. Rice, and Robert S. Fish. Life and Death of New Technology: Task, Utility and Social influences on the Use of a Communication Medium. In *CSCW'94 proceedings*, pp. 13–21, Oct. 1994.
- [24] Paul P. Jovanis and Anthony Moore. Perception of new electronic media for business communication. *Transportation*, Vol. 14, pp. 329–344, 1988.
- [25] Robert M. Krauss. The role of the listener: Addressee influences on message formulation. *Journal of Language and Social Psychology*, Vol. 6, No. 2, pp. 81–98, 1987.
- [26] Robert M. Krauss and Susan R. Fussell. Other-relatedness in language processing: Discussion and comments. *Journal of Language and Social Psychology*, Vol. 7, pp. 263–279, 1988.
- [27] Carl F. Graumann and C. Michael Sommer. Perspective structure in language production and comprehension. *Journal of Language and Social Psychology*, Vol. 7, pp. 193–212, 1988.
- [28] Herbert H. Clark and Edward F. Schaefer. Contributing to discourse. *Cognitive Science*, Vol. 13, No. 2, pp. 259–294, 1989.
- [29] John C. McCarthy, Victoria C. Miles, and Andrew F. Monk. An experimental study of common ground in text-based communication. In *CHI'91 proceedings*, pp. 209–215, 1991.
- [30] Robert M. Krauss and Susan R. Fussell. Mutual Knowledge and Communicative Effectiveness. In *Intellectual Teamwork: Social and technological Foundations of Cooperative Work*, chapter 5, pp. 111–145. Lawrence Erlbaum Associates, 1990.
- [31] Rachel Reichman. Conversational coherency. *Cognitive Science*, Vol. 2, No. 4, pp. 283–327, 1978.
- [32] John C. McCarthy, Peter C. Wright, and Andrew F. Monk. Coherence in text-based electronic conferencing: Coupling text and context. *Journal of Language and Social Psychology*, Vol. 11, No. 4, pp. 267–277, 1992.

- [33] John C. McCarthy and Andrew F. Monk. Measuring the quality of computer-mediated communication. *BEHAVIOUR & INFORMATION TECHNOLOGY*, Vol. 13, No. 5, pp. 311–319, 1994.
- [34] Abigail J. Sellen. Speech patterns in video-mediated conversations. In *CHI'92 proceedings*, pp. 49–59, May 1992.
- [35] Andrew Monk, John McCarthy, Leon Watts, and Owen Daly-Jones. Measures of Process. In Peter Thomas, editor, *CSCW Requirements and Evaluation*, chapter 9, pp. 125–139. Springer, 1996.
- [36] Abigail J. Sellen. Remote conversations: The effects of mediating talk with technology. *Journal Of Human-Computer Interaction*, Vol. 10, pp. 401–444, 1995.
- [37] Jolene Galegher and Robert E. Kraut. Computer-Mediated Communication for Intellectual Teamwork: A Field Experiment in Group Writing” CSCW'90. In *CSCW'90 proceedings*, pp. 65–78, Oct. 1990.
- [38] J. H. Erik Andriessen. The Why, How and What to Evaluate of Interaction Technology: A Review and Proposed Integration. In Peter Thomas, editor, *CSCW Requirements and Evaluation*, chapter 8, pp. 107–124. Springer, 1996.
- [39] 井上智雄, 岡田謙一, 松下温. テレビ番組のカメラワークの知識に基づいた tv 会議システム. *情報処理学会論文誌*, Vol. 37, No. 11, pp. 2095–2103, 1996.
- [40] Stephen Reder et al. The Temporal Structure of Cooperative Activity. In *CSCW 90 Proceedings*, pp. 303–316, 1990.
- [41] 溝口徹夫. SE の要求仕様獲得の実際 - 要求仕様決定の要因と課題. *情報処理*, Vol. 33, No. 6, pp. 617–619, June 1992.
- [42] Bashar Nuseibeh, Jeff Kramer, and Anthony Finkelstein. Expressing the relationships between multiple views in requirements specification. In *15th international conference on Software Engineering*, pp. 187–196, May 1993.
- [43] Daniel P. Freedman and Gerald M. Weinberg. *HANDBOOK OF WALK-THROUGHS, INSPECTIONS, AND TECHNICAL REVIEWS*. Little, Brown and Company (Inc.), 1982.
- [44] Elliot Soloway and Kate Ehrlich. Empirical studies of programming knowledge. *IEEE Transactions on Software Engineering*, Vol. SE-10, No. 10, pp. 595–609, 1984.
- [45] Lucy M. Berlin. Consultants and apprentices: Observations about learning and collaborative problem solving. In *CSCW92 Proceedings*, pp. 130–137, Nov. 1992.
- [46] Beth Adelson and Elliot Soloway. The role of domain experience in software design. *IEEE Transactions on Software Engineering*, Vol. SE-11, No. 11, pp. 1351–1360, 1985.
- [47] 垂水浩幸. グループウェアのソフトウェア開発への応用. *情報処理*, Vol. 33, No. 1, pp. 22–31, Jan. 1992.
- [48] Bill Curtis. Implication from empirical studies of the software design process. In *Proc. of Int. Conf. by IPSJ to Commemorate the 30th Anniversary*, pp. 127–134. IPSJ, 1990.

- [49] Motoshi Saeki. Communication, Collaboration and Cooperation in Software Development - How Should We Support Group Work in Software Development? In *Proceeding of 1995 Asia Pacific Software Engineering Conference*, pp. 12-20, Brisbane, Australia, Dec. 1995. Australiaon Computer Society, IEEE Computer Society Press.
- [50] Clarence Ellis and Jacques Wainer. A Conceptual Model of Groupware. In *CSCW'94 Proceedings*, pp. 79-88, 1994.
- [51] M. Saeki, S. Sureerat, and K. Yoshida. Supporting Distributed Individual Work in Cooperative Specification Development. In *In Lecture Notes in Computer Science(CISMOD'95)*, 1995.
- [52] Motoshi Saeki, Kinji Matsumura, Jun'ichi Shimoda, and Haruhiko Kaiya. Structuring utterance records for requirements elicitation meetings based on speech act theory. In *Proc. of 2nd International Conference on Requirements Engineering (ICRE'96)*, 1996.
- [53] Colin Potts, Kenji Takahashi, and Annie I. Anton. Inquiry-based requirements analysis. *IEEE Software*, Vol. 11, No. 2, pp. 21-32, Mar. 1994.
- [54] Mark Keil and Erran Carmel. Customer-Developer Links in Software Development. *Commun. ACM*, Vol. 38, No. 5, pp. 33-44, May 1995.
- [55] Salah Bendifallah and Walt Scacchi. Work Structures and shifts: An Empirical Analysis of Software Specification Teamwork. In *ICSE 11th proceedings*, pp. 260-270. IEEE Computer Society and ACM SIGSOFT, 1989.
- [56] Haruhiko Kaiya, Motoshi Saeki, and Koichiro Ochimizu. Design of a Hyper Media Tool to support Requirements Elicitation Meetings. In Hausi A. Müller and Ronald J. Norman, editors, *CASE'95: Proceedings Seventh International Workshop on Computer-Aided Software Engineering*, pp. 250-259, Toronto, Ontario, Canada, Jul. 1995. IWCASE, IEEE Computer Society Press.
- [57] G. M. Schneider, Johnny Martin, and W. T. Tsai. An experimental study of fault detection in user requirements documents. *ACM Transactions on Software Engineering and Methodology*, Vol. 1, No. 2, pp. 188-204, Apr. 1992.
- [58] 桑名栄二. ソフトウェア履歴利用の研究動向. 電子情報通信学会誌, Vol. 77, No. 5, pp. 531-538, May 1994.
- [59] Jeff Conklin and Michael L. Begeman. gIBIS: A hypertext tool for exploratory policy discussion. In *CSCW'86 Proceedings*, Dec. 1986.
- [60] Jintae Lee and Kum-Yew Lai. What's in Design Rationale. *HUMAN-COMPUTER INTERACTION*, Vol. 6, No. 3 & 4, pp. 251-280, 1991.
- [61] Bill Curtis, Herb Krasner, Vincent Shen, and Neil Iscoe. On Building Software Process Models Under the Lamppost. In *ICSE 9th proceedings*, pp. 96-103. IEEE Computer Society and ACM SIGSOFT, 1987.
- [62] 中小路久美代. 非同期協調作業における意図とコンテキストの役割. 情報処理学会研究報告 96-GW-20, pp. 19-24, Oct. 1996. グループウェア.

- [63] T. W. Malone, K. R. Grant, K. Y. Lai, R. Rao, and D. Rosenblitt.
- [64] 竹田尚彦ほか. カード操作ツールにおけるソフトウェア開発における 要求定義工程の支援. 第2回発想支援ツール・シンポジウム論文集, 1993.
- [65] Donald C. Gause and Gerald M. Weinberg. *Exploring Requirements: Quality Before Design*. Dorset House Publishing Co., Inc., 353 West 12th Street, New York, NY 10014, 1989.
- [66] L. Brothers et al. ICICLE: Groupware For Code Inspection. In *CSCW90 Proceedings*, pp. 169-181, Oct. 1990.
- [67] Steve Whittaker and Heinrich Schwarz. Back To The Future: Pen And Paper Technology Supports Complex Group Coordination. In *CHI'95 proceedings*, pp. 495-502, May 1995.
- [68] Laurent Karsenty. An Empirical Evaluation of Design Rationale Documents. In *CHI96 Proceedings*, pp. 150-156, Vancouver, BC Canada, Apr. 1996.
- [69] K. C. B. Yakemovic and E. J. Conklin. Report on a Development Project Use of an Issue-Based Information System. In *CSCW 90 Proceedings*, pp. 105-118, 1990.
- [70] Jonathan Grudin. Why cscw application fail: Problems in the design and evaluation of organizational interfaces. In *CSCW 88 Proceedings*, pp. 85-93, 1988.
- [71] 村越広享, 海谷治彦, 落水浩一郎. ネットワークを介した共同作業における阻害要因の分析. 日本ソフトウェア科学会第12回大会論文集, pp. 233-236, Sep. 1995.
- [72] Hiroyuki Murakoshi, Haruhiko Kaiya, and Koichiro Ochimizu. An Analysis of Obstruction in Cooperative Work over a Computer Network. In *International CASE Symposium '95 (CICS'95)*, pp. 98-103, Oct. 1995.
- [73] 村越広享, 海谷治彦, 落水浩一郎, 佐伯元司. 非同期型のコミュニケーションを用いた共同作業における 阻害要因の分析. 情報処理学会研究報告 96-SE-109, Vol. 96, No. 41, pp. 25-32, May 1996. ソフトウェア工学.
- [74] 村越広享, 海谷治彦, 落水浩一郎. 共同ソフトウェア開発における非同期型コミュニケーションの生産物への影響. 佐伯元司, 権藤克彦 (編), レクチャーノート/ソフトウェア学17 ソフトウェア工学の基礎 III, pp. 166-169, 会津若松, Dec. 1996. 日本ソフトウェア科学会 FOSE'96, 近代科学社.
- [75] 落水浩一郎, 門脇千恵, 藤枝和宏, 堀雅和. ソフトウェア分散開発支援環境「自在」のアーキテクチャ設計. 信学技報 SS94-18, Vol. 94, No. 135, pp. 1-8, Jul. 1994. ソフトウェアサイエンス.
- [76] 落水浩一郎. ネットワークを介した共同作業の支援環境構築にむけて. JAIST Reserch Report IS-RR-97-0017S, 北陸先端科学技術大学院大学, 石川県能美郡辰口町旭台 1-1, Apr. 1997.
- [77] 門脇千恵, 落水浩一郎. 非同期分散型会議の事象駆動型討議プロセスによるモデル化と調整支援への応用. 情報処理学会研究報告 94-SE-96, Vol. 94, No. 6, pp. 193-200, Jan. 1994.

- [78] 近野章二, 落水浩一郎. グループウェアベース「栞」を用いた電子会議内容の進捗状況と文書化の支援. 情報処理学会研究報告 96-SE-107, Vol. 107, No. 12, pp. 89-96, Jan. 1996.