

## ソフトウェア設計用 グループウェアに関する一考察

垂水浩幸

日本電気(株) 関西C & C研究所

多くの場合ソフトウェア開発チームは地理的・時間的に分散している。分散した環境ではそうでない環境と比較してチーム内部のコミュニケーションが円滑に行なわれず、その結果はソフトウェアの生産性や品質に影響する。本稿では分散環境におけるソフトウェア開発にコミュニケーションの立場からアプローチし、分散した環境がコミュニケーションにどのような影響をもたらすかについて述べる。次にコミュニケーション媒体の特性とソフトウェア開発の作業の特性を比較し、ソフトウェア開発にグループウェアを導入する場合の指針と課題について述べる。

## ON GROUPWARE TECHNOLOGIES AND SOFTWARE DESIGN

Hiroyuki Tarumi

NEC Kansai C&C Research Lab.

A software development team is usually distributed geographically or has internal time difference. Such distributed environment brings about lack of communication and low productivity or quality. From the viewpoint of human communication, this paper discusses the relationship between distribution and quality. Guidelines and problems in introducing groupware into software development activity is also discussed.

## 1 はじめに

ソフトウェアの開発は分散して行なわれるのが普通である。特に最近では人材を地方や海外に求める傾向が現れてきており、分散する範囲はますます広がってきている。地理的な分散の他にも、フレックスタイム、深夜残業、海外との時差など時間的な分散の問題も存在する。分散開発は一箇所で行なわれる開発よりも効率が悪いのは当然であるが、非技術的な理由による制約により、さらに分散せざるを得ない状況になってきている。

ソフトウェア開発はグループ協同作業の代表的なものであり、グループウェアの応用分野としては有望と考えられる。本稿では、グループウェアをソフトウェア開発に応用する方向性を見出すことを目的とし、ソフトウェア開発作業におけるコミュニケーション、およびコミュニケーション媒体の特性について考察する。合わせてソフトウェア開発におけるコミュニケーションの重要性をソフトウェア品質の立場から述べ、分散した環境でコミュニケーションが低下する理由を考察し、コミュニケーション向上のための方策を探る。

## 2 分散のもたらす障害

### 2.1 コミュニケーションとソフトウェア品質

分散のもたらす弊害は、結局コミュニケーションの有効量(頻度×質)を下げることでであると換言できよう。頻度と質については後に述べるが、ここでは、ソフトウェア開発のために有効なコミュニケーションをコミュニケーションの有効量と非形式的に定義しておく。

コミュニケーションの有効量が下がれば、その結果以下のような症状が現れ、ソフトウェアの質や生産性が下がることになる。

- さまざまな認識のずれ、勘違い。
- 仕様伝達、変更伝達などの遅れ。
- 質問事項解決の遅れ。
- データや資料収集に要する時間の増大。
- 命令管理系統の遅れ。

この中で、例えば「認識のずれ」は要求定義などの上流工程でよくみられる現象であり、コミュニケーションの低下は認識のずれの解決(すなわち上流工程の進捗)を遅らせることになる。また、設計工程などでは設計上の判断を行なうために資料の収集や他人への質問を行なっていく必要があり、それが遅ればすなわち進捗が遅れる<sup>1</sup>。いずれの場合も、単に遅れるだけではなく、多くの場合はスケジュール上の制約により遅れを取り戻さないまま次の工程に進んだり、情報収集不足のまま設計上

<sup>1</sup>システム開発者の作業の $\frac{1}{4}$ 以上が質問であるという報告もある [7]

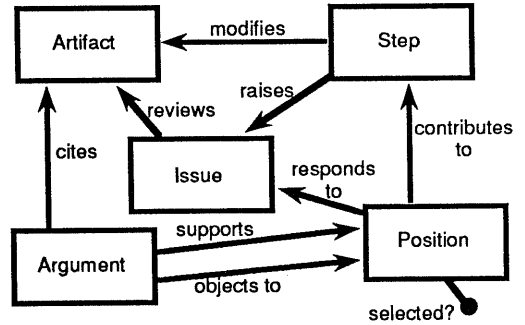


図 1: Potts のモデル

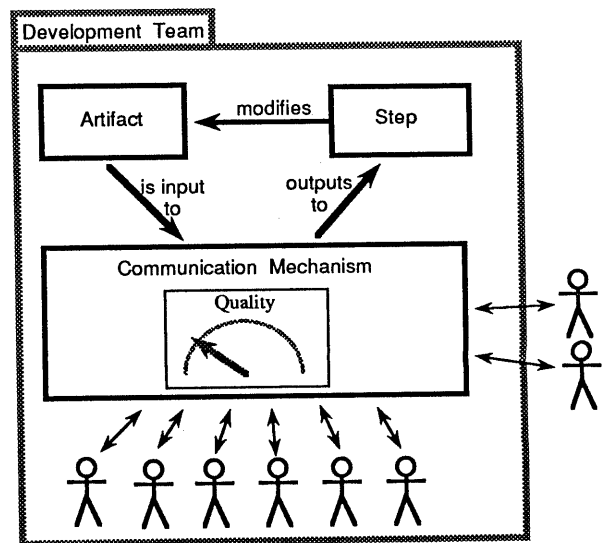


図 2: Potts モデルの一般化

の判断を下すようなことになる。これらは下流で後戻り工程を生む原因となり、すなわちソフトウェアの質の低下につながるようになる。

これをもう少し細かく考えてみる。ソフトウェア開発作業は成果物 (artifact) から別の成果物を得る過程の繰り返しであり、コミュニケーションはその各過程で必要に応じてとっていかなければならない。これを表す一つのモデルが Potts [11] によって示されている (図 1)。Potts は、成果物から成果物への書き換え作業 (step) を形成するための議論を argument, issue, position で表現した。これは IBIS という議論モデル [2] によっている。

これを品質の立場で考えると、成果物  $A_m$  から成果物  $A_n$  を得る作業において十分なコミュニケーションをとらなければ  $A_n$  の品質は上がらないということである。ただし、ここではコミュニケーションとは議論だけ

でなくあらゆる情報交換を包含するものとする。そのことは、Potts のモデルを特定の議論モデルに依存しないよう一般化することにより、図 2 で示すことができよう。図中の Communication Mechanism は議論モデルを一般化したもので、その中の 'Quality' はコミュニケーションが次段の成果物の品質に影響することを示している。これは次の仮説によっている。

$$Q_n = \frac{c_1}{c_0} f(A_m) + g(A_m) + h$$

- $Q_n$  : 成果物  $A_n$  の品質  
 $c_0$  :  $A_m$  から  $A_n$  を得るのに必要なコミュニケーションの総量  
 $c_1$  : 実際に行なわれたコミュニケーションの有効量 ( $c_1 \leq c_0$ )  
 $f, g, h$  : 品質を決定する未知の関数。

$f, g, h$  のうち  $f, g$  は  $A_m$  に依存するとしたが、これはもちろん  $A_m$  の品質  $Q_m$  にも依存するということである。また  $f, g, h$  は他の様々なものに依存するはずであるが、ここでは未知の関数ということにしておき深くは追求しない。 $c_0$  は  $A_m$ 、 $A_n$  (の理想的な形)、作業者の持っている知識などに依存しているはずである

さて、コミュニケーションの観点から品質をあげるには  $c_0$  を小さくするか、 $c_1$  を大きくすればよい。

$c_0$  を小さくする技法には例えばモジュール化技法やオブジェクト指向などがある。これらは、プロジェクト担当者のグループの間の独立性を高めて、コミュニケーションの必要量を減らす。例えば  $A_m$  があるモジュールの基本設計仕様書、 $A_n$  が詳細設計仕様書であるとすると、 $A_m$  が独立性の高いモジュールなら  $A_n$  を作るために他のモジュールの仕様書を参照する回数が減少するはずであるから、 $c_0$  を減らしたことになる。

また、スキルの高い要員を集めることも  $c_0$  を小さくするのに役立つ。なぜならば教育のために必要不可欠なコミュニケーション量が小さくなるからである。 $A_m$  から  $A_n$  を導く担当者のスキルが高ければ、その人は他人に質問することも少ないだろうし、回りから教えるべきことも少ないだろう。

$c_1$  を大きくするためには管理者が気を配って意志疎通を円滑にしたり、効果的に会議を開いたりすることが有効である。また、仕様書のフォーマットを統一したり、有効と言われているレビュー技法<sup>2</sup>を用いるなど、コミュニケーションの質 (= ここでの意味は、適切な情報を相手に確実に伝えること — 後述) を向上させて短時間に必要な情報を確実に伝えるように工夫することも大事だろう。

その他にコミュニケーション支援ツールを用いたり、分散の形態を工夫することでコミュニケーションを局所

<sup>2</sup>例えば、[3] には、ライフサイクルの上流で用いるものから下流で用いるものまで、数多くのレビュー技法が紹介されている。

化してコミュニケーションコストを下げることにより、 $c_1$  を大きくすることができるはずである。これらが分散ソフトウェア開発の研究課題と言える。

さて、コミュニケーションの有効量は頻度と質の積であると定義したが、それらと分散との関係について考察してみる。

## 2.2 コミュニケーション有効量の低下要因

### コミュニケーションの頻度

コミュニケーションの頻度を下げる要因には、物理的障害、心理的障害がある。

物理的障害とは伝達媒体自体の制限によるものや、媒体が利用できない物理的理由がある場合などで、例えば、地理的分散が原因で面談や公式対面会議ができないこと、電子メールの遅延による応答性の悪さ、多忙なスケジュールを原因とするコミュニケーション頻度の低下、時差を原因とする電話時間の制限などである。

心理的障害とは、伝達媒体自体の障害ではなく、その媒体を利用する意欲をそぐものである。例えば出張手続き、多忙な相手からコミュニケーション時間を得ることに対する後ろめたさ、気難しい相手とのコミュニケーションの煩わしさ、使いにくいユーザインタフェースを持つコミュニケーションツールなどである。

分散環境においては、物理的障害だけでなく心理的障害も発生することに注意すべきであろう。例えば電話一つかけるにしても、相手が忙しいかもしれないし、不在かもしれない。このような細かな煩わしさがコミュニケーションの頻度を下げる可能性がある。同じビルの顔の見える範囲にいれば、相手の暇そうな時を狙って声をかけられるので、分散している場合よりも有利である。

心理的障害には個人差がある。相手に遠慮なく電話をかけるのが平気な人は、心理的障害をあまり感じず、離れた相手とコミュニケーションをとるのもそれほど不得手としない。また、他人に伝えたり質問したりする用件をこまめにメモして確実に消化していくようなタイプの人には、心理的障害の影響は比較的小さい。分散環境がもたらす影響には個人差がある。

### コミュニケーションの質

コミュニケーションの質の定量化は困難であるので、ここでは定性的に述べる。

ソフトウェア開発における質の高いコミュニケーションとは、

開発のために必要な適切な情報を、適切なタイミングで適切な相手に確実に伝えること

と定義できるだろう。

適切な情報とは、例えば次のようなものである。

- 仕様記者に対しては記述の誤りの情報

- 管理者に対しては進捗情報
- 未熟な担当者に対しては適切な教育
- ユーザに対しては、そのユーザが最も求めている要求を開発中のシステムがどの程度満足させるのか ... という情報

次に、適切なタイミングとは、例えば障害情報をできるだけ早く伝えることである。

確実に伝えるとは、正確に伝えること、必要ならば記録を残すことその他、会議中相手が眠そうな顔をしているときに重要事項を小声で言わないとか、特に重要な情報は文書だけではなく対面で相手の目を見ながら真剣に話して重要さを肌で理解させる、といった配慮まで含む。

### 3 コミュニケーションの特性

#### 3.1 コミュニケーション媒体の特性

さて、コミュニケーションの質を確保するためには、適切なコミュニケーション媒体を用いなければならない。例えば、特に正確さの求められる情報を送るには電話よりも FAX や 電子メールが適しているし、重要な命令を伝えるには対面で迫力を示すのが最も適している。また、上流工程では議論を活発にするために、リアルタイム性や発展性のある媒体が望ましい。すなわち、コミュニケーション媒体にはそれぞれ特性があり、その特性に応じて適切に利用されなければならない。

ここでは、媒体の特性の指標となる項目を挙げ、それらがどの媒体に含まれ、あるいは含まれていないかを示す。

**正確さ** 誤りを含まないこと、相手の意識に確実に伝達されること。<sup>3</sup>

**記録性** 記録が残ること。

**大量性** 一度に大量の情報を伝達できること

**引用性** 他から引用した情報を伝えられること。

**リアルタイム性** リアルタイムで相手に伝わり、反応が返ること。

**発展性** 話題が発展的に広がること。

**時間任意性** 参加者が都合のよい時間にコミュニケーションに従事できること。

**非文字情報** 文字以外の図、イメージなどを伝えられること。

**非言語情報** 気迫、表情、身振り手ぶりなどを伝えられること。

<sup>3</sup>伝達媒体やそれを運営する環境によらず、発信者自身の誤りや、情報発信能力の不足を原因とする正確さの欠如もあるが、それらはこの研究の範囲外であるのでここでは考慮しない。

この他に一斉同報性やセキュリティなどの特性が考えられるが、ここでは言及しない。

表1に、各伝達媒体の持つコミュニケーションの特性を示す。ここで、面談は非公式の予約なしで行なわれる対面コミュニケーションを、TV会議は音声と疑似動画の使用できるTV会議室を、在席電子会議は自席ワークステーションを用いた分散会議を、電子メールは主にテキストベースのものを考えている。ただし電子メールは必ず相手に読まれるものと仮定しており、そうでなければ正確さの評価はもう少し落ちる。

#### 3.2 ソフトウェアライフサイクルとコミュニケーション特性

次に、ソフトウェア開発の各工程で求められているコミュニケーション媒体の特性について考察してみる。それを表2に示す。なお、ここでは命令などの管理情報の伝達は除いている。

**要求定義** プレインストーミングや KJ 法などの創造技法を用いるため、ここでは特にリアルタイム性・発展性のある媒体が要求される。また、言語として書き留めるまでに図で示すことが多いので非文字情報を伝達する必要がある。すぐに言葉にはなりにくいユーザの意図をつかむためには、非言語情報を伝達する必要がある。

**設計** 各種チャートを用いて仕様書を記述するため、非文字情報を取り扱う必要がある。設計段階以降では、上流で作成した文書や、既存の文書を適宜引用していくため、引用性がある方が望ましい。また、これより下流では文書に記述できない非言語情報は意味をなさなくなってくる。また、ここより下流にいくと細かな問題が頻繁に発生し、それらを地道に解決していく必要があるので時間任意性のある媒体が望ましい。

**コーディング・単体テスト** これより下流では、創造力はあまり必要としないので、発展性は不要になる。反面、文書が大量になってくるので、大量性が求められる。

**結合テスト** 作業がパッチ的になるので、コミュニケーションの頻度は少し下がり、時間任意性はあまり必要ではなくなる。反面、テストデータが大量に流れるため、大量性はますます必要になる。

**保守** ユーザの要求の強さや障害の緊急性を伝えるため、非言語情報を伝達する必要性が再び現れる。緊急な場合には時間任意性とリアルタイム性が必要である。

表1、表2より、ソフトウェアライフサイクルの各工程について、それぞれに適応するコミュニケーション媒体がわかる。その代表的なものを表3に示す。

表 1: コミュニケーション媒体の特性

	面談	資料付面談	公式対面会議	電話	TV会議	在席電子会議	電子メール	Fax
正確さ	×	△	△	×	△	○	○	○
記録性	×	△	△	×	×	○	○	○
大量性	×	×	×	×	△	○	○	△
引用性	×	△	△	×	×	○	○	△
リアルタイム性	○	○	○	○	○	○	×	×
発展性	○	○	○	○	○	△	×	×
時間任意性	×	×	×	△	×	△	○	○
非文字情報	×	○	○	×	△	△	△	○
非言語情報	○	○	○	△	△	△	×	×

○: 強い  
 △: 弱い  
 ×: なし

表 2: ソフトウェア開発工程の要求するコミュニケーション媒体特性

	要求定義	設計	コーディング 単体テスト	結合 テスト	保守
正確さ	○	○	○	○	○
記録性	○	○	○	○	○
大量性	×	△	○	○	○
引用性	△	○	○	○	○
リアルタイム性	○	○	△	△	○
発展性	○	△	×	×	×
時間任意性	△	○	○	△	○
非文字情報	○	○	△	△	△
非言語情報	○	△	×	×	△

○: 必要  
 △: 場合により必要  
 ×: 不要

表 3: 開発の各工程で望ましいコミュニケーション媒体

要求定義	面談、資料付面談、公式対面会議
設計	在席電子会議 + 電子メール、資料付面談 + 電子メール
コーディング 単体テスト	電子メール
結合テスト	電子メール
保守	在席電子会議

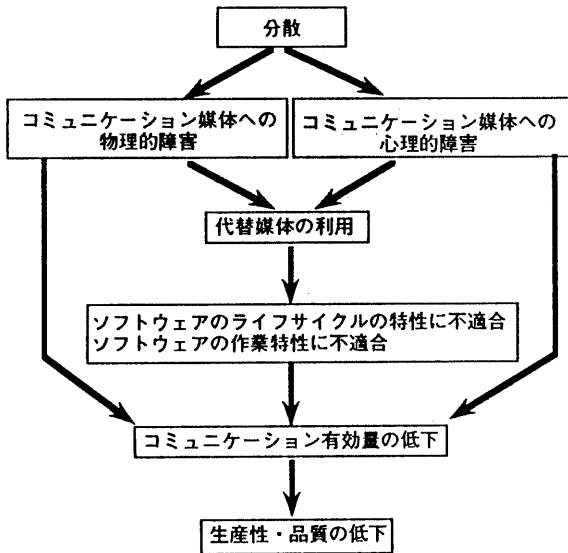


図 3: 分散とソフトウェア生産性・品質との関係

## 4 解決の方向と課題

### 4.1 コミュニケーション媒体の進化

これまで述べてきたように、分散はコミュニケーション媒体の自由な利用を物理的・心理的に制限し、その結果他のコミュニケーション媒体の代替利用を余儀なくさせる。ソフトウェアのライフサイクル・作業特性に不適合なコミュニケーション媒体の利用はコミュニケーションの有効量を下げ、最終的にはソフトウェアの生産性・品質を落すことになる(図3)。

この因果関係を断ち切るには、次のような手段が考えられる。

- 既存のコミュニケーション媒体を改良するか、または他の媒体で置き換え、分散時の心理的障害を軽減する。例えば、
  - 電話は相手の邪魔になるので電子メールを利用する。
  - 会議室予約の事務手続きを簡素化する。
- 既存のコミュニケーション媒体を改良し、あるいは新しいコミュニケーション媒体を開発し、分散した状態でもソフトウェア開発作業に有効な特性が出るようにする。例えば、
  - 対面の隣場感を持つ、新しい分散会議システム。
  - Fax に引用性を持たせるために計算機と結合し、電子的に記録された文書をそのまま送れるようにする。

- 電子メールでは非文字情報が伝わりにくいので、マルチメディア電子メールに改良する。

すなわち、分散という前提のもとでソフトウェアの生産性と品質を向上させるには、既存のコミュニケーション媒体を改善したり、あるいは新しいコミュニケーション媒体を導入したりすることが有効であると言える。それがソフトウェア開発用のグループウェアの基盤になる。

### 4.2 ソフトウェア開発作業の特徴

グループウェアをソフトウェア開発作業に応用するに当たって、ソフトウェア作業におけるコミュニケーションの特性について整理しておく必要がある。ソフトウェア開発作業を、それ以外の一般の作業と比較すると、経験上次のような特性があると考えられ[13]、これらは太字で示したようなコミュニケーション媒体の特性を要求する。

- 1) 参照すべき資料が大量で、検索が困難である。[引用性、大量性]
- 2) 詳細で正確な内容を伝達しなければならない。[記録性、引用性]
- 3) 高度で知的なコミュニケーションが必要であり、参加者の疲労などによる集中力の欠如はコミュニケーションの結果に悪い影響をもたらす。[時間任意性]
- 4) 各種チャートや表などの構造的データを利用する。[非文字情報]
- 5) 開発作業への参加者が多様でそれぞれ役割があり、専門が異なる。また、ライフサイクルの全体に関わる人もいれば、一部にしか関わらない人もいる。
- 6) 作業標準が組織によってまちまちである。
- 7) 作業が長期間に及び、途中での担当員の交替も珍しくない。[記録性]
- 8) 他人の修正、仕様の変更、日程の変更など、予期できない変化が多い。[記録性]
- 9) 他人の担当部分が理解しにくく、作業の属人性が高い。[記録性、時間任意性]

さらに、1) は検索への知的支援、使いやすいユーザインタフェースを、5) はライフサイクルや参加者の役割に応じた機能を、5), 6) はツールのカスタマイズビリティを、8) は(急な変更に対する)柔軟性およびリボジトリとの連結機能を要求する。

これらの特性要求と解決手段を表4に示す。

本節で示したようなソフトウェア開発作業らしい特徴は特に設計工程以降によく現れる。従って設計工程以降はソフトウェア専用のグループウェアがあれば特に有効である。

表 4: ソフトウェア開発支援への課題と解決手段

課題	解決手段
引用性	電子メール、在席電子会議
大量性	電子メール、在席電子会議
記録性	電子メール、在席電子会議、 Fax
非同期性	電子メール、 Fax
知的支援	(今後の課題)
ユーザインタフェース	(今後の課題)
ライフサイクルに応じた支援	表 3参照
参加者の役割に応じた支援	(今後の課題)
カスタマイザビリティ	(今後の課題)
柔軟性	(今後の課題)
リポジトリとの連結	(今後の課題)
非文字情報	例えば鼎 [12] の応用

### 4.3 電子メールの有効性

既存のコミュニケーション媒体の中では、特に電子メールに注目したい。表 3、表 4からわかるように、ソフトウェア開発という作業に対して広く応用でき、効果が大きいことが予想される。それらを以下にまとめる。

電子メールは非同期媒体であるため、以下の利点がある。

- 議論に用いる場合、参加者は自分の都合と体調の良い時間に参加できる。このことは、先に述べた「高度で知的なコミュニケーション」を実現するのに都合がよい。
- 同期媒体と異なり、全員参加できることが保証されているので、属人性が高いというソフトウェアの特性から言っても、議論には都合がよい。
- 時間任意性が高い上、媒体利用に対する心理的障害が極めて小さい。このことは、議論の種を埋もれさせないことにつながり、コミュニケーションの有効量を増大させる。

また、電子媒体であるため、以下の利点がある。

- 引用性が高く、ソースコードや仕様書を議論の題材にするには好都合である。
- 記録が残る。このことの利点は [14] で述べられている通り。

さらに、電子メールには

- 誰でも簡単に使える。
- ソフトウェア開発だけに限らず、応用範囲が広い。日常利用が可能。

などの特性があり [1][4]、同じ非同期のグループウェア用電子媒体である“ハイパーテキスト共有”(gIBIS など)と比べて、普及の可能性が格段に大きい。グループウェアはグループの大多数のメンバに受け入れられ、積極的に利用されなければ役に立たない [6][9] ので、ハイパーテキスト共有のような形態はよほどレベルの高いグループでなければ受け入れられないであろう。また、普及のためには、導入決定権を持つ管理者にとって有効でなければならない [5] ので、日常利用可能であることは大きな普及推進力である。

反面、現状の電子メールには

- A) 重要なメールとそうでないメールの判別がしにくい。また必ず読んでもらえるとは限らない。
- B) 整理が困難。
- C) 議論が発散する。
- D) 発言者が偏る。
- E) 他ツールとの連絡が困難。

などの欠点があり、これらは改善していく必要がある。

A) のメールのフィルタリングに関しては、Object Lens [8]、I-CEM [10] などの研究例がある。B) は表 4で示したように今後の課題である。C) と D) を解決するには議論の方法論とともにその方法論を具現するグループウェアを提供しなければならない。IBIS などは方法論の一例であるが、ソフトウェア開発(特に設計)をターゲットとしたものではないので、現場でそのまま使用できるかどうかの見極めが重要である。E) は表 4の「リポジトリとの連結」と同じことで、今後の課題である。

また電子メールが普及しているという米国での報告 [1] に比して、日本ではキーボードに慣れない人が残っており、グループ全体へツールを導入する時の障害にな

る可能性がある。このように、文化的な背景についても考慮していく必要がある。

## 5 おわりに

本稿では、ソフトウェア開発をコミュニケーションの立場から見直し、ソフトウェア開発にグループウェアを導入することの有益性を導いた。また、導入のための指針の一端を示した。

## 謝辞

熱心に議論していただき、適切な助言をいただいた日本電気(株) 関西 C & C 研究所の真名垣所長、宮井課長に感謝します。

## 参考文献

- [1] Bullen, C. V. and Bennet, J. L. : Learning From User Experience with Groupware, *CSCW'90 Proceedings*, ACM (1990), pp.291-302
- [2] Conklin, J., and Begeman, M. L. : gIBIS: A Hypertext Tool for Exploratory Policy Discussion, *CSCW'88 Proceedings*, ACM (1988), pp.140-152
- [3] Freedman, D. P., Weinberg, G. M. : *HANDBOOK OF WALKTHROUGHS, INSPECTIONS, AND TECHNICAL REVIEWS, Evaluating Programs, Projects, and Products (Third Edition)*, Little, Brown, and Company, Inc., 1977, (岡田他監訳、TBS 出版会)
- [4] Galegher, J. and Krout, R. E. : Computer-Mediated Communication for Intellectual Teamwork: A Field Experiment in Group Writing, *CSCW'90 Proceedings*, ACM, pp.65-78, (1990)
- [5] Grudin, J.: Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces, *CSCW'88 Proceedings*, ACM (1988), pp.85-93
- [6] 石井裕: CSCW: コンピュータを用いたグループワーク支援の研究動向、コンピュータソフトウェア、Vol.8, No.2 (1991)
- [7] Kedzierski, B. J. : Communication and Management Support in System Development Environments, in *Computer Supported Cooperative Work: A Book of Reading*, Greif, I., Ed., Morgan Kaufmann (1988)
- [8] Lai, Kum-Yew and Malone, T. W. : Object Lens: A "Spreadsheet" for Cooperative Work, *CSCW'88 Proceedings*, ACM (1988), pp.115-124
- [9] Markus, M. L. and Connolly, T. : Why CSCW Applications Fail: Problems in the Adoption of Interdependent Work Tools, *CSCW'90 Proceedings*, ACM (1990), pp.371-380
- [10] 松浦宣彦、平岩真一、松下温: 情報の効果的活用に基づいたグループウェア、情報処理学会マルチメディア通信と分散処理研究会、50-14、(1991)
- [11] Potts, C. : A Generic Model for Representing Design Methods, *Proc. of the 11th International Conference on Software Engineering*, IEEE (1989), pp.217-226
- [12] 暦本、垂水、菅井、他: エディタを部品としたユーザインタフェース構築基盤: 鼎、情報処理、第31巻第5号、(1990)
- [13] 垂水浩幸: グループウェアとソフトウェア開発、日本ソフトウェア科学会ソフトウェア工学研究会(関西地区)、SW-91-8-1、(1991)
- [14] Yakemovic, K. C. B. and Conklin, E. J. : Report on a Development Project Use of an Issue-Based Information System, *CSCW'90 Proceedings*, ACM (1990), pp.105-118