ソフトウェアの仕様化過程における協調作業のモデル化

海谷治彦　　佐伯元司

東京工業大学

ソフトウェア仕様化 / 設計過程における協調作業のモデルを構築するために行なった実験についての研究報告を行なう．一般にソフトウェア開発では複数の人間が作業に携わるため，協調作業としての人間の作業活動を明らかにする必要がある．我々はグループによるソフトウェア仕様化 / 設計作業を分析 / 評価する観点から重要であると思われる性質にはどのようなものがあるかを調べるために，作業の模擬実験をビデオカメラを用いて記録し，発話記録の分析を行なった．さらに，その実験結果を基に，作業活動の分析のためのモデルを構築した．

VDP: Framework for Analyzing Cooperative Software Process

Haruhiko Kaiya　　Motoshi Saeki

Tokyo Institute of Technology

This paper presents empirical studies for modeling cooperative software design process. We should clarify human activities as cooperative work because software development process is cooperation of many workers. We construct a model for analysing cooperative software process to recorded several pre-designed meeting by using videotape and to analyze their verbal description.

1　はじめに

我々は，ソフトウェア開発，特に仕様化 / 設計段階においての協調作業を支援する手法を研究している [6]．

協調作業の支援を行なうには，支援対象の特徴を的確に捉え，何を使ってどのように支援するかを決定する必要がある．そのためには，ある観点から見て重要であると考えられる対象の性質を抽出し，それらを記述するための枠組，すなわちモデルを定義することが重要である．

協調作業のモデルとして，Winograd らの Conversation Model [3], [4], [5] や，Conklin らの IBIS Model [1] などが提案されているが，これらのモデルは一般的すぎて実際の作業に自然な形で適用し，作業支援を行なうことは難しい．そのため，仕様化 / 設計過程固有の協調作業を分析しモデルを構築することが必要である．

我々は，実際のソフトウェア開発の模擬実験を行ない，それを分析することで知見を得る実験的手法を用いて，人間の思考過程にあったモデルの構築をめざす．その理由は，現在までにソフトウェアの仕様化 / 設計過程を特徴付ける性質は明らかにされていないため，これらを明らかにするために実際の作業活動を観察し，分析を行なう必要があるからである．しかし，このような分析手法も確立していないため，本研究では，まず分析可能な対象の性質を選び記述する作業，すなわち分析する観点からのモデル化の議論を行なう．

まず初めに，グループにおける仕様化作業では，作業における発話 (Verbal Description) の中に意味ある性質が含まれているという仮定の基に，模擬実験の様子のビデオ記録のプロトコル解析を行なった．

その分析結果から，いくつかの性質が分析のため抽出可能であることを結論し，それを基に，ソフトウェア開発における協調作業を分析する観点からモデル化した．

適切な構文で表現できないが重要である性質は，計算機による支援が困難であると考えられるので，非形式的な記述にとどめておく．

2　実験指針 / 分析指針

実験 / 分析の目的は，協調作業支援を考慮しつつ，実際のソフトウェア開発の模擬実験を分析するための枠組である分析モデルを構築することである．

そのために，どのような指針で実験 / 分析を行なったかについて述べる．

2.1　実験設定

我々が支援対象としているソフトウェアの仕様化 / 設計作業の模擬実験の環境設定の指針として，以下の 2 条件が考えられる．

1. 実際の仕様化 / 設計作業に近い環境
2. 協調作業を行なうのに理想的な環境

これらの条件を考慮すると，

　１つの与えられた問題を全員で対話しながら解決していく作業形態

が望ましい．よって実験は以下のような環境設定の下で行なった．

作業者: 5 ~ 6 人．
　(理由) 会議において，局所的な会話が起こりにくい範囲の人数．

作業者の役割: 与える．
　(理由) ソフトウェア開発では，立場や専門分野の違った作業者の協調作業が常であるから．
　(役割の例) 対象としているソフトウェアにおける作業分担に即した役割．顧客，利用者，製作者，調整者などの立場の違いによる役割．リーダー，メンバーなどの身分差による役割．熟練者，初心者などの熟練度の違いによる役割．

作業場所: 対面式円卓会議，黒板メモなどの使用が可能．
　(理由) 作業者間の緊密なコミュニケーションが可能である．

記録方法: １台のビデオカメラによる発話者中心の記録．
　(理由) 作業中に分析のための記録をさせる方法とは異なり，作業者は仕様化作業に専念できる．インタビューなどを後から行なうのとは異なり，正確で客観的な情報を記録できる．

以上の設定より，作業者間で局所的な対話が並行列的に生じるような形態は扱わない．

2.2　分析指針

我々が実験記録から客観的に抽出できるのは，それぞれの作業者の発話の順序列である．

我々はそれぞれの発話を適当な視点から分類し，その分類の出現順序の特徴や，頻度を調べることで，分析モデルの構造を決定していく．視点の例として，どの作業者が発話を行なったかという基準で分類する作業者による視点などが考えられる．

3　実験方法 / 分析方法

実験指針に従って，2 種類の実験を行なった．それぞれの実験は数回の会議から成り立っている．我々は以下の仮定を基に，実験１，２での役割を作業者に与えた．

　対象システムの実現上の作業分担が仕様化作業において重要な要素である
　顧客，利用者，製作者，調整者など作業に関わる立場が重要な要素である

実験 1
　作業者に対する役割として，システム実現上の作業分担を行なうように，作業者に対して事前に指示する．実際に製作するソフトウェアは，論文などの図を描くためのグラフィックエディタであり，このシステムの要求を作業者に対して与える役割を持っている顧客が１名参加している．

実験 2
　実験 1 で得た知見より，作業者に対する役割を，システム開発の立場に応じて，作業前に事前に割り当てる．詳細は次章で述べる．
　設計したソフトウェアはハイパーテキスト型の設計過程分析ツールであり，そのツールの詳細は [7] で報告されている．

分析を行なうために，ビデオカメラによる記録を計算機上のテキストファイルに書き起こし，分析の対象である発話に対して，図 1に示すような情報を付加し，整理する．

@　<視点 1>　<視点 2>　<視点 3>　· · ·

図 1: テキスト化した発話記録の付加情報

図 2に，テキスト化した発話記録の例を示す．この例では<視点 1>は作業者名，<視点 2>は発話開始時刻，<視点 3>は役割の視点におけるある分類が記述される．*は分類不能であることを示す．

```
@ W     3.40    D
それで，とりあえず領域をきめて，
そのあと書くのはねえ，こまっているのですよ．
とりあえず，このへん
%画面の図の領域の四角の先頭
にカーソルか何かを出さなければならないのですよね．
でもそんなことないかな ...
@ W     4.00    U
下の方にコマンドラインを作って，ピとかいくの？
@ O     *       D
そう
@ I     *       D
それがらくではないですか？
@ W     *       U
どっちにしようかな
@ O     4.20    U
カーソルだと見にくいから ...
@ K     4.25    D
図に差し込む程度なんでしょう？
@ I     *       D
だから下に出てきて，
だから最初に枠を決めるのではなくて，
下に打ち込んだ大きさからフォントから枠の大きさを計算して，
@ W     4.35    D
ああ，そうか
文字だけ適当にいれておいて，領域だけ書いて，ここ
%画面上の位置
には，なんにもださない．
@ I     4.40    D
いやいや，領域じゃなくて．
```

図 2: テキスト化した発話記録の例

4　実験結果と分析

実験結果を可視化するために，縦軸に発話の発生順序を上から下に，横軸に発話を視点での分類をとりプロットした順序列グラフを作成する．例えば，視点として発話者名を選ぶことで，作業の最も客観的なデータを作成することができる．

次に，ある発話から次の発話へ移る時の視点上の分類の遷移を視点遷移表にまとめる．表の縦軸を遷移前の分類を示し，横軸を遷移後の分類を示し，表の要素は，その遷移が起こった回数を示している．例えば，視点として作業分担を選んだ結果，特定の分担から特定の分担への遷移が多い場合，その 2 つの分担は結び付きが強いことが結論できる．

4.1　実験 1

実験 1 は合計 3 回の作業を表 1に示すような時間で行なった．

	第 1 回	第 2 回	第 3 回
作業時間	2:12	1:15	1:16

表 1: 実験 1 の作業時間

4.1.1　作業者

図 3に，発話の視点を，どの作業者が発話したかで分類した順序グラフを示す．

作業は要求を与える顧客を含めて，5 人の作業者で行なった．それぞれの作業者には，第 1 回終了後，表 2に示すような作業分担が行なわれた．ここでの分担作業は，次回の会議開催日までにソフトウェアの特定の部分の試案を作る役割を決定した作業であり，以後の作業 (第 2,3 回) では，その試案について全員が議論する形で仕様化が行なわれた．

作業者	作業分担
作業者 1	入出力
作業者 2	画面
作業者 3	文字
作業者 4	編集
作業者 5	描画

表 2: 作業者の作業分担

4.1.2　作業分担

発話の視点をどの作業分担に言及しているかで分類した順序列グラフを図 4に示す．第 1 回目の会議では存在しなかった利用者のマウス入力を一括して扱う分担 (マウス) が作業者の提案によって第 2 回後半で発生するが，議論の結果，画面の分担に吸収される．

次に作業分担を視点として選んだ視点遷移表を表 3に示す．例えば，第 2 回では画面について言及している発話の次の発話は画面の分担自身に言及しているものが一番多く，93 遷移を占めている．

ある分担に言及した発話の直後に行なわれる発話は，その分担自身に言及しているものが一番多い．よって，作業分担を順序列上の発話列を 1 つの塊として捉えることが可能である．

4.1.3　作業者 / 分担

図 3と図 4を比較してみる．

作業者 2 は分担前は比較的少ない発話数にも関わらず，第 2 ，3 回は発話数がかなり増えている．逆に作業者 5 は第 1 回の発話頻度に比べ，第 3 ，4 回の発話頻度はかなり減っている．

4.1.4　発話の連続性

異なる分担に言及している発話でも，密接な関連があり，相互に影響しあっていることが考えられる．そこで，発話間の関連度合いを連続性という観点から整理してみる．

発話記録のそれぞれの発話に対して，直前の発話に影響を受けているものを連続，そうでないものに断絶という性質を付けることで連続性という視点を定義する．例えば，質問に対する反論，意見に対する反論や質問などは連続の分類される．図 5に連続性の順序列グラフを示す．

断絶は時間経過にしたがって減少している．

4.1.5　実験 2 への知見

実験 1 の観察の結果より，最初の仮定における立場とはどのようなものであるかを整理した．その立場を表 4に示

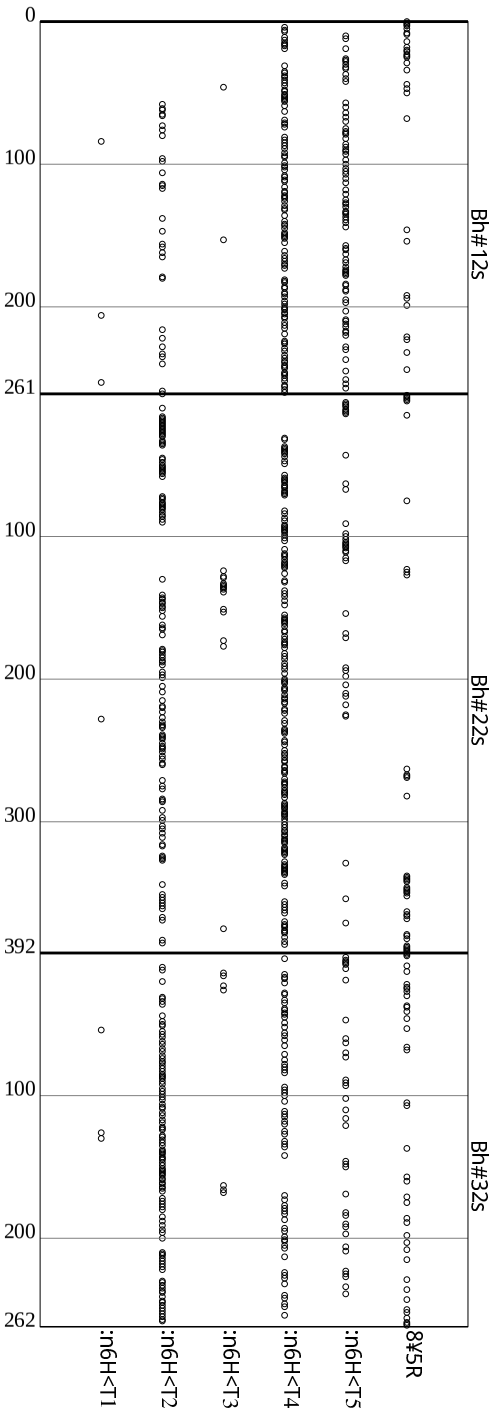

図 3: 実験 1 の作業者による順序グラフ

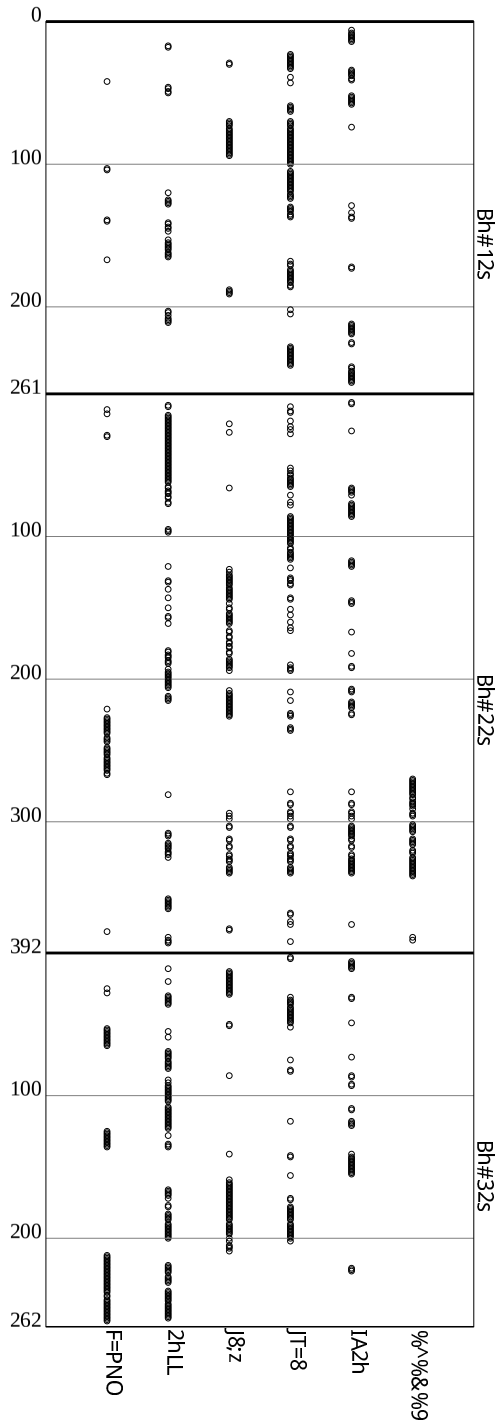

図 4: 実験 1 の分担による順序グラフ

第 1 回

	入出力	画面	文字	編集	描画	マウス
入出力	2	1	0	[3]	0	0
画面	0	[20]	0	2	2	0
文字	0	0	[26]	25	1	0
編集	0	4	24	[88]	8	0
描画	2	0	1	6	[36]	0
マウス	0	0	0	0	0	0

第 2 回

	入出力	画面	文字	編集	描画	マウス
入出力	[30]	3	1	4	0	0
画面	2	[93]	26	32	17	4
文字	3	22	[61]	28	26	14
編集	6	29	28	[56]	24	18
描画	1	15	24	22	[49]	23
マウス	0	9	14	19	24	[41]

第 3 回

	入出力	画面	文字	編集	描画	マウス
入出力	[65]	34	2	0	3	0
画面	33	[89]	21	25	13	0
文字	2	23	[52]	22	1	0
編集	1	21	20	[37]	5	0
描画	3	15	1	3	[26]	0
マウス	0	0	0	0	0	0

表 3: 作業分担の視点遷移表

立場	内容	略記
顧客	作業過程に対する要求．	K
利用者	ソフトウェアに対する機能の要求．	U
製作者	ソフトウェアに対する要求の具体化．	D
調整者	作業過程の具体化．	C
書記	決定事項の記録，確認．	S

表 4: ソフトウェア開発における立場

し，その例を図 6に示す．立場の視点は図の第 3 カラム目であり，表の略記で記述している．

顧客と利用者は要求を出す立場にあり，調整者と技術者は要求を加工，実行する立場にある．別の見方をすると，顧客と調整者は作業過程に対して言及するのに対し，利用者と技術者は対象ソフトウェアに言及する立場である．

4.2 実験 2

実験 2 は合計 4 回の実験を表 5に示すような時間で行なった．

	第 1 回	第 2 回	第 3 回	第 4 回
作業時間	2:45	2:27	2:54	3:30

表 5: 実験 2 の作業時間

4.2.1 作業者 / 立場

実験 1 の整理を基に，それぞれの作業者に対して，表 4 に示すような立場に基づいた役割を与えた．

図 7に発話をどの作業者が発話したかで分類した順序グラフを示し，図 8に発話をどの立場に立って発話が行なわれたかで分類した順序グラフを示す．実験 1 の分担のグラフと違って，順序列上の特定分類の偏在が見られない．また，割り当てられた立場を越えた発話がいくつかあった．

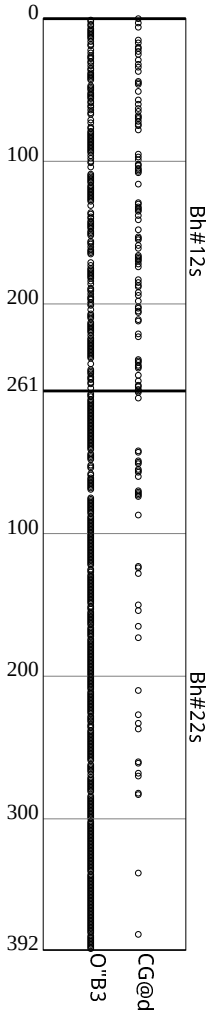

図 5: 実験 1 の連続性による順序グラフ (第 3 回は省略)

次に，立場の視点の遷移に着目した視点遷移表を表 6に示す．

4.2.2 立場毎の内部状態

立場の視点からの順序グラフから特別な特徴がみられないため，§4.1.4で述べた連続性の視点を利用し，それぞれの立場毎に，ある発話が後に続く発話や話題に対してどの様な影響を与えているかで分類する影響視点を考える．

さらに，ある発話に影響されることで，他者に対して発話を行ない状態を変化させるような内部状態を立場毎に想定する．

以上の視点を立場毎に順序機械で表現すると，図 10のような例がみられた．

4.3 視点の分類

それぞれの発話は，外部的な特徴と対応付けるか，時間順序の上で前に行なわれた発話との関係を用いるか，後に

```
@ O     *       S       G
draw 系でメニューを決めるんですか？
@ H     *       C       G
黒板にイメージを書いてみてよ
@ I     *       *       *
% 黒板に o_6 o_7 の図を書く
%（メニューの出方の図）
@ I     *       D       G
o_6 o_7 どちらの方法が良いか
@ H     *       U       G
o_6 o_7 共に使えた方が良い
@ I     *       C       *
あとは機能さえ割り当てれば終り
%（機能の図 A）
@ I     *       U       D
%「直線，円」
@ I     *       U       D
連続直線は別に設けなくていいの？
@ I     *       U       D
%「軌線」
@ I     *       U       D
%「長方形，多角形，塗りつぶし」
@ I     *       U       D
お客さんは論文の挿絵に使いたいと言うので「矢印」
@ K     *       D       D
矢印は直線に対する修飾だよ
@ W     *       S       D
%「矢印」を消す
@ I     *       U       E
%「拡大縮小」
@ K     *       D       E
軌跡とかは図形でしょ 拡大縮小は別に書いた方がいい
@ W     *       S       E
%「拡大縮小」を書き直す
@ W     *       U       E
%「回転，コピー，移動，グルーピング」
@ I     *       D       E
グルーピングをしてから拡大縮小とかするからグルーピング
は機能として上だな
```

図 6: 立場による作業例

行なわれた発話との関係を用いるかで視点を変えてきた．

例えば，話題と役割は第 1 の種類の視点に入り，連続性は第 2 の種類の視点に入り，影響は第 3 の種類の視点に入る．

これは [2] における 3 purpose of content analysis research design に対応している．

5 協調作業モデル: VDP

実験 1 より，発話を言及している作業分担 (つまりソフトウェアの部分) を基にまとめて捉えることができる．この発話のまとまりを話題とする．

実験 2 より，作業者は顧客，利用者，調整者，技術者などの立場にたって，話題に取り組んでいると捉えることができる．このような立場に従って分類した発話の集合を役割とする．

また，それぞれの役割に含まれる発話を特定の話題に与えている影響で分類し，それぞれの発話を入出力とした順序機械を役割毎に想定する．このような立場毎の内部状態を基にした順序機械をプロトコルとする．

以上のような視点を基に，ソフトウェアの仕様化 / 設計過程を分析するための枠組である VDP モデル (Verbal Description Process Model) を提案する．

モデルは話題，役割，プロトコルの 3 つの視点を基に構

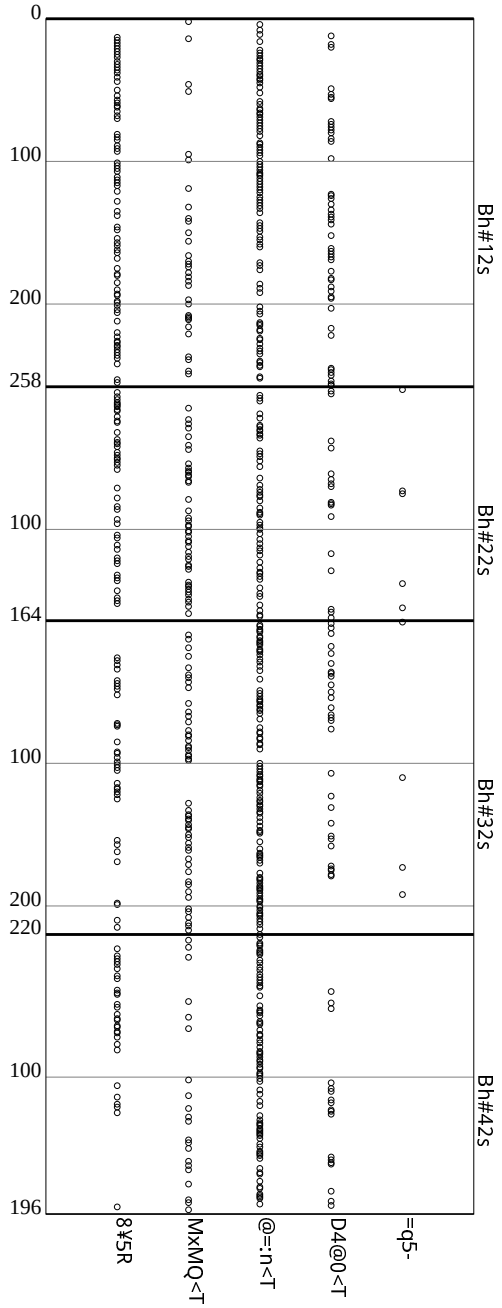

図 7: 実験 2 の作業者による順序グラフ

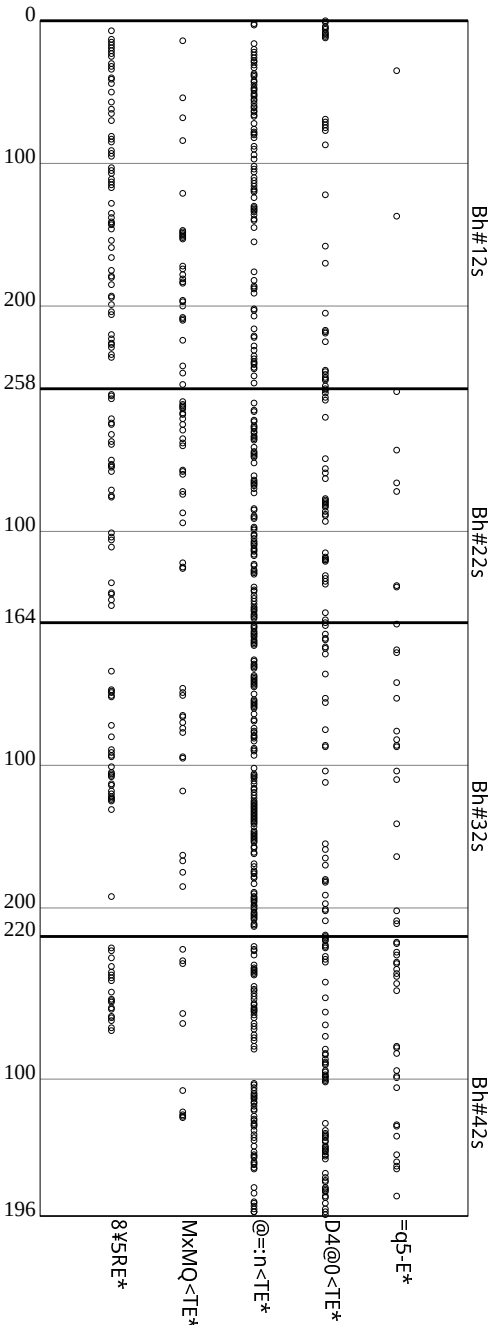

図 8: 実験 2 の役割による順序グラフ

第 1 回

	顧客的	利用者的	製作者的	調整者的	書記的
顧客的	6	6	31	3	1
利用者的	6	10	3	5	0
製作者的	31	7	29	6	1
調整者的	7	1	6	12	0
書記的	1	0	0	0	0

第 2 回

	顧客的	利用者的	製作者的	調整者的	書記的
顧客的	6	4	10	6	1
利用者的	3	6	11	2	0
製作者的	11	6	39	7	1
調整者的	5	7	6	11	2
書記的	0	0	4	1	1

第 3 回

	顧客的	利用者的	製作者的	調整者的	書記的
顧客的	11	3	15	1	1
利用者的	2	2	3	1	1
製作者的	13	9	82	15	6
調整者的	3	0	15	6	5
書記的	1	1	9	4	1

第 4 回

	顧客的	利用者的	製作者的	調整者的	書記的
顧客的	4	1	9	2	3
利用者的	1	2	5	2	1
製作者的	11	3	27	13	11
調整者的	3	4	13	28	7
書記的	3	1	9	10	5

表 6: 立場の視点遷移表

成されており，それぞれの視点を Topic view, Role view, Protocol view とする．

以上の 3 つの view は図 9に示すような関係がある．

仕様化過程全体は話題の時間順序的な連接として表現される．

それぞれの話題はいくつかの役割によって議論が行なわれる．

また，それぞれの役割はその役割に特有なプロトコルを持ち，入出力となる発話はその時点で注目しいる話題に影響を与え，場合によっては次の話題に議論を移すこともできる．

例えば，図 9では，ある役割が B という発話を話題に対して行ない，それに対して，他の役割が B を受けて C という発話を行なうことで，現在の話題を連続させていることが表現されている．

実際のプロトコル上での発話の種類は，図 10に示すような，誘導，合意，提案，意見，説明などが考えられる．

6 おわりに

我々は実験的手法を用いて，ソフトウェアの仕様化 / 設計過程を分析するための枠組となるモデル VDP を提案した．

今後の課題として，それぞれの視点の性質の評価，判断基準を確立する必要がある．例えば，

- 同様の話題が断片的に議論されることをどの様に評価すべきか？

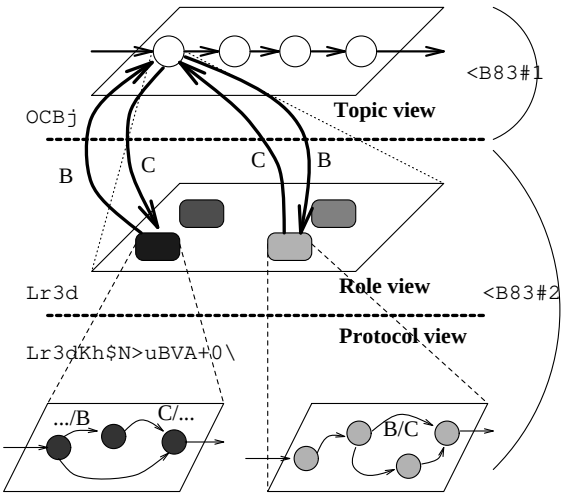

図 9: モデルの図

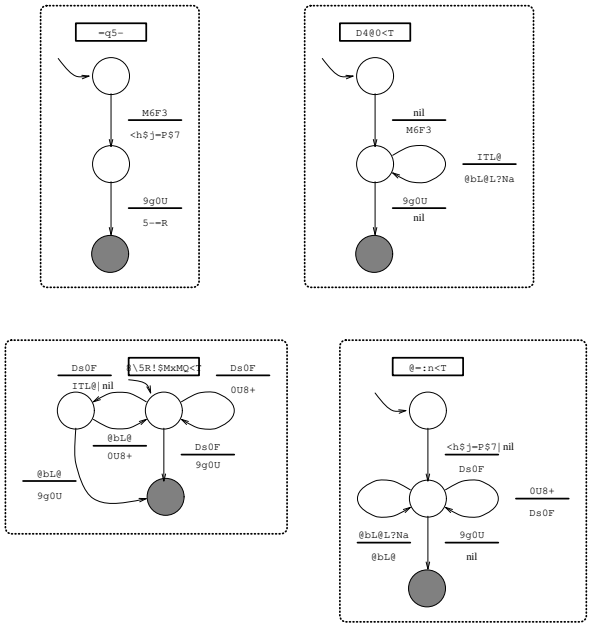

図 10: Protocol view の例

- すべての役割の主張の影響の及ぼし具合をどのように判断すべきか，偏りがあったほうがいいのか，影響が均等におよぼしているほうが良いのか？
- 役割毎のプロトコルをどの様に評価するか？プロトコルの最適解は存在するか？
- 話題の時系列的な並びが，対象とするソフトウェアの構造と何らかの関係があるのか，ないのか？

などが考えられる．

また，今回の枠組では見えにくい点として，

- アイディアのとりこぼしがないか．
- グループのサイズがどのように影響するか．

などが考えられる．

今後の実験への知見として，

- 役割の feasibility を与えるための，教育的実験の必要性．

が挙げられる．

参 考 文 献

[1] Jeff Conklin and Michael L. Begeman. gIBIS: A hypertext tool for exploratory policy discussion. In *CSCW'86 Proceedings*, Dec 1986.

[2] OLE R. HOLSTI. *Content Analysis for the Social Science and Humanities*. Addison-Wesley, 1969.

[3] Terry Winograd. A language perspective on the design of coopearative work. In *CSCW'86 Proceedings*, Dec 1986.

[4] Terry Winograd. Where the action is. *BYTE*, Vol. 13, No. 13, Dec 1988.

[5] Terry Winograd and Fernand Flores. *Understanding Computers and Cognition*. Ablex Publishing Corporation, Norwood, N.J., 1986.

[6] 海谷, 佐伯. ソフトウェアの仕様化過程における協調作業の分析. 情報処理学会ソフトウェア工学研究会, pp. 77–4, Feb 1991.

[7] 渡辺智弘, 佐伯元司. 思考過程モデルに基づいたツールを用いた仕様化作業の分析. 電子情報通信学会全国大会, p. 6:43, Oct 1990.