ソフトウェアの仕様化作業における会話構造の分析

ANALYSIS OF CONVERSATION STRUCTURE IN DEVELOPING SOFTWARE SPECIFICATIONS

海谷 治彦　　佐伯 元司
Haruhiko KAIYA　　**Motoshi SAEKI**
東京工業大学 工学部
Tokyo Institute of Technology

1　はじめに

実際のソフトウェア仕様化/設計過程の記録をとることは, その過程で得られた仕様書を理解する際の助けになるという利点がある. さらに種々の記録を集め, 分析することにより,

- 過程中の作業段階の難易度といった評価尺度.
- 仕様書などの成果物の品質に関する評価尺度.

を求め, 過程の改善や成果物の品質改善に役立てることができると思われる.

本研究では, この仕様化/設計過程の記録を分析し, 仕様書の品質に関する評価尺度を設定し, 実際の過程の記録の評価を行なった.

仕様化/設計過程には数多くの作業形態がある中で, 本研究では, 複数の作業者が会議を通して仕様決定を行なう形態を分析対象とした. なぜなら, その作業形態が仕様化/設計過程の中で重要な役割を果たし, また, 観測できる中間情報を数多く含んでいるからである. 会議における会話では, さまざまな話題が提案され, 何度も繰り返し議論されるものもあれば, 提案されたまま顧みられないものもある. 数多く議論されている話題を含む仕様書の部分は高い品質を持つと予想される. なぜなら, そのような箇所は, 仕様書を設計する過程において, 多方面から検証され, 多くの対案を出しながら検討されているからである. よって, 会議の発話記録より, 仕様書中のどの部分が多く議論されていたかを表す評価尺度を設定し, 仕様書との関連を調べた.

ソフトウェア設計過程を記録することのできる計算機上のツールはいくつか開発されている [1], [2], [3], [4]. しかし, それらは, 特定の設計方針に基盤をおいた作業方法を作業者に行なわせており, 作業者自身の設計スタイルが阻害されている可能性がある. そこで, 我々は, 個々の作業者がソフトウェアの仕様化/設計作業を行なうのに, できる限り作業者自身の思考や作業を保持できるように記録する方法を選び, その記録に対する分析を行なった. つまり, 本研究ではツールを用いて記録を行なう手法ではなく, ビデオカメラを用いて記録を行なう手法を採用した. 我々と同様に, ソフトウェア開発における作業者自身の設計スタイルを重視した形で記録をとり, その分析を行なった事例もいくつか存在する [5], [6] が, 本研究の目的とは異なっている.

我々は, ソフトウェア設計過程を分析するための枠組としての研究を続けており, それには, 個人作業を対象としたもの [7] や, グループ作業を対象としたもの [8, 9] がある. この報告は, その研究の一貫である.

第 2章では, 我々の分析方法の目的と, その手続きを説明する. 第 3章では, 我々の方法を実験的に行なわれたソフトウェア開発の一部に適応した事例を紹介する. そして, 第 4章で, この方法の利点, 欠点などの議論を行ない, そして最後に, 今後の研究方向について述べる.

2　分析の方法

2.1　分析対象の条件

我々がどのような作業形態を取り上げ, それをどのように記録し. その記録のどの部分に注目したかについて説明する.

- どのような作業形態に注目するか?
  実際のソフトウェア開発作業は, いろいろな作業形態を含んでいる. 例えば, アイディアを考えるためや, 考えをまとめるための個人作業や, 対面式, もしくは非対面式で行なう情報交換や, 意見調整などの形態である [10]. 我々は, 分析対象として, 作業者が対面して会話を行なう形態の作業 (以後, これを会議を呼ぶ) を選んだ. このような形態の作業は, 分析者が観測できる中間情報が多いため, 開発過程に生じる有益な情報を観測できる可能性が高い.
- どのようにして記録するか?
  分析対象となる作業の記録は, できる限り, 作業者の作業に負担をかけない方法で行なわなければならない. そのために, 特定の作業記録をするためのツールや方法を利用することは避け, 作業者の発話と行為をビデオカメラを用いて記録する方法を採用した. この方法を用いることで, 分析者は繰り返し作業の観察を行なうことができる.
- どのような情報に注目するか?
  我々の採用した方法から得られる記録は, あまりにも多くの情報を含んでいる. そこで, それらの記録から, 必要な情報だけを抽出しなければならない. 我々は, 記録された行為の時間的な量の情報よりも, 順序の情報を重視する.

2.2 会話の構造

このような性質を抽出するためのデータ収集方法として, 実際のソフトウェア開発過程における発話や行為をビデオカメラを用いて記録するのが最善の方法であると考えた.

一般に, 仕様作成過程では, いろいろな仕様書内における事柄が決定される. 我々はそのような事柄を話題と呼ぶことにする. 仕様化作業における会話では, 作業者が話題に対して何らかの行為を行なうことで, 1 つ 1 つの話題は決定されて行く. このような行為の内で, 作業者が初めて話題の存在を認識する行為を提案と呼び, その提案に対して何らかの結論を与える行為を言明とする. さらに, すでに言明されている話題について確認を行なう行為を確認と呼ぶことにする. 我々は仕様化作業における仕様書の品質の尺度は, これらの行為に依存すると考える.

一般に, 1 つの会議の間には, 仕様書のある部分に言及している会話の断片が複数の箇所に点在している. この断片を会話間隔 (conversation interval) と呼び, 以後, **CI** と略記する. 1 つの CI の中にはいくつかの提案, 言明, 確認に相当する行為が含まれている.

図 1は, 仕様書のある部分に言及している CI のある会話内の分布と, それぞれの CI に含まれる行為の数を示している. 図中の円は, ある行為を示し, 直線で話題と結ばれている行為は提案を表し, 破線で話題と結ばれている行為は言明を表し, 点線で話題と結ばれている行為は確認を表している. 例えば, 図中で最初に出現する CI は, 話題 1 に対する 1 つの提案を含むが, 最後の CI は, 2 つの確認 (話題 2, 話題 3) と, 1 つの提案 (話題 4) を含む.

我々は, CI 毎の新しい話題の提案と, すでに提案されている話題に対する言明と参照を行なっている行為の回数を数え, それから算出される値を基に, 対応する仕様書の部分の評価を行なう. その値は, 直観的には, 何度も繰り返し議論される話題が多ければ, 高い値を示すような値であり, 我々はその値を参照頻度と呼ぶことにする. 参照頻度は以下のように定義される:

$$\text{参照頻度} = \frac{\text{言明} + \text{確認}}{\text{提案} + \text{言明} + \text{確認}}$$

我々は会話に現れる CI を最終的な仕様書の部分に対応づけることで分類を行なう. この分類に使用するカテゴリを仕様カテゴリ (specification category) と呼ぶことにし, 以後, SC と略記する. 我々は会話の持つ順序情報の内で, ある SC に分類された一連の CI の持つ参照頻度の変化に注目する. もし, 一連の CI の参照頻度が, CI の会話の中における出現順序に従って値が高くなるとすれば, その SC は高い品質を持つと我々は判断する.

図 2を用いて, CI, SC とについての直観的な説明を行なう. 図 2は, 縦軸に SC をとり, 横軸に時間をとっている. 図中の円は 1 つの行為をしめしており, 番号の同じものは同じ行為を示している. すなわち, 2 番の行為のように, 複数の SC に属する行為がありうる. 例えば, 第 3章で取り上げる, 「グループによるスケジュール管理システム」の仕様化過程において, 次のような発話が行なわれたとする.

「定期的な行事も,
グループ共有の行事もコピーする.」

この発話は,

1. 定期行事
2. グループ共有行事

の双方の SC に属し, 話題として,

行事の保存方法.

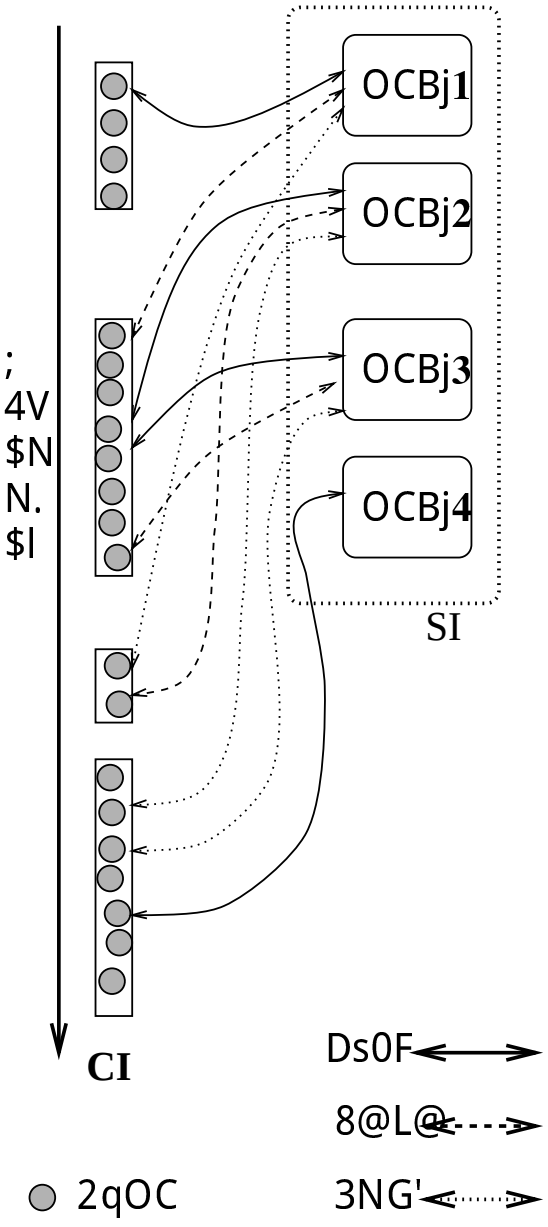


図 1: CI における提案, 言明, 確認

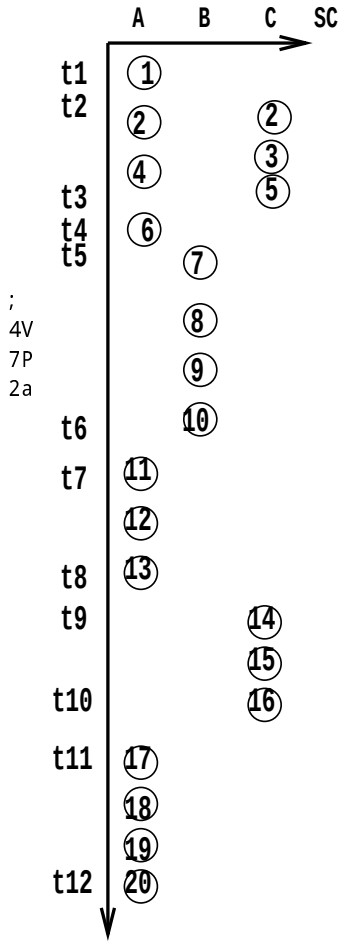


図 2: ある会話の構造

に対する言明に当たる. 時間 t1 から t4, t7 から t8, t11 から t12 の 3 つの時間間隔は, A という SC に属する一連の CI である. その他の CI も, 同様に, どれかの SC に分類される. ある話題の提案が行為 3 でおこなわれた場合, その話題に対する言明, 確認は行為 5, 14, 15, 16... 以外では行なわれない.

3 分析方法の適用事例

この章では, 1991 年に行なわれたソフトウェア開発と, その分析について紹介する. このソフトウェア開発は, 分析を行なうために実験として設定された側面を持つ.

3.1 実験の概要

この開発は, 実験としての側面以外に, 情報処理系の短大の学生が, 卒業研究に作成するプログラムを作成するための側面を持っている. 我々が分析対象とする部分は数回の会議からなりっており, 4 種類の作業者が, その会議に参加した. 表 1は, その 4 種類の作業者の種類と, 従事した人数と, その作業内容が示されている.

この実験では 9 回の会議が行なわれており, 図 3に, 行なわれた月日と, 会議時間を示す. 会議は 2 種類行なわれており, 図中の中黒の矩形で表されている会議は全ての作業者が参加して行なわれたものであり, 中白の矩形は設計者

表 1: 作業者の種類

種類	人数	役割
発注者	1	ソフトウェアに対する要求を与える
プログラマ	4 ~ 5	実装する
調整者	1	作業者間の調整をとる
設計者	2	仕様書を作る

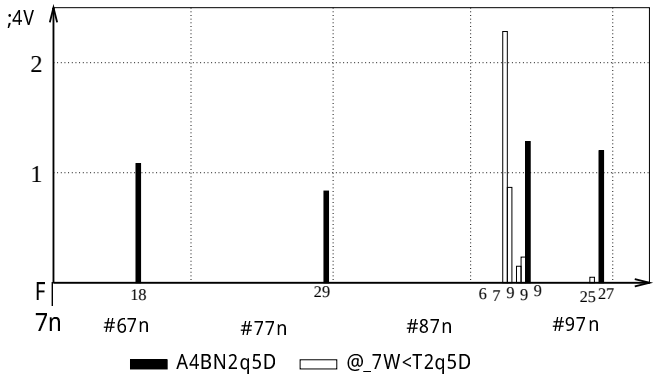


図 3: 9 回の会議の開催日と時間

のみで行なわれた会議である. 前者の会議を全体会議と呼び, 後者を設計者会議とよぶ.

最後の全体会議に, 設計者から 15 ページの仕様書が提出され, この実験における設計段階は終了する. ここで提出された仕様書を最終仕様書と呼ことにする.

このプロセスで開発を行なったソフトウェアは,「グループによるスケジュール管理システム」と言う. このシステムは, 複数の利用者と, その利用者の中のグループのスケジュールを効率的に管理するソフトウェアである.

3.2 分析結果

我々は設計者会議のみに注目し, 最終仕様書から 7 つの SC を選んだ. 全体会議についての分析は文献 [11] で, 設計終了後の実現段階については文献 [9] で報告されている. 表 2にそれぞれの SC の内容を示す.

図 4に, 分析対象となった会議に出現した CI を示す. この図は対象となった 4 回の設計者会議に対応する 4 つの時刻グラフから成り立っている. それぞれのグラフは, 縦軸に上から下へ会議の経過時間をとり, 横軸には表 2で示した SC をとっている. グラフ中の中黒の矩形が 1 つの CI を示している. 例えば, SC3(利用者管理のカテゴリ) に分類される CI は, 9 月 6 日の会議のグラフの 12 分から 16 分と, 115 分から 121 分の 2 箇所に存在する. 縦軸の一番右に分類さ

表 2: 仕様カテゴリ

カテゴリ名	内容
SC1.	メールシステム
SC2.	システム開始時の利用者界面
SC3.	利用者管理
SC4.	グループに共有されるスケジュール
SC5.	利用者の操作
SC6.	スケジュールのデータ構造
SC7.	定期的なスケジュール

表 3: SC1 の参照頻度

No.	date	提案	言明	確認	参照頻度
1	9 月 6 日	6	0	0	0.00
2	9 月 6 日	2	1	2	0.60
3	9 月 6 日	3	1	4	**0.63**
4	9 月 6 日	7	3	3	0.46
5	9 月 7 日	3	0	3	**0.50**
6	9 月 7 日	1	0	0	0.00
7	9 月 7 日	0	0	1	**1.00**
8	9 月 7 日	2	0	5	0.71
total		24	5	18	0.49

表 4: SC2 の参照頻度

No.	date	提案	言明	確認	参照頻度
1	9 月 6 日	1	0	0	0.00
total		1	0	0	0.00

れる CI は, どの SC にも分類されなかった CI を示している. 例えば, 9 月 9 日の第 1 回と第 2 回と 9 月 25 日の会議はほとんど表 2 のカテゴリでは分類されなかった. 分類できなかった会話の内容を調べると, 作業分担や全体会議での発表の手順などについての話し合いであった.

表 3, 4, 5, 6, 7, 8, 9 は, それぞれ, SC 1, 2, 3, 4, 5, 6, 7 に属する CI の参照頻度の変化と, それぞれの SC における 3 つの行為の累計から計算した参照頻度を total として表示してある. 例えば, 表 3は SC1 に属する 8 つの CI それぞれの参照頻度が示されており, 直後の CI の参照頻度よりも高い値の参照頻度を持つ CI が 3 箇所存在する. このように, ある SC に分類される CI の中で, 時間的に直後に出現する CI の参照頻度より高い参照頻度を持つ CI をピークと呼ぶ. 我々は唯一のピークを持つ SC を高品質と考える.

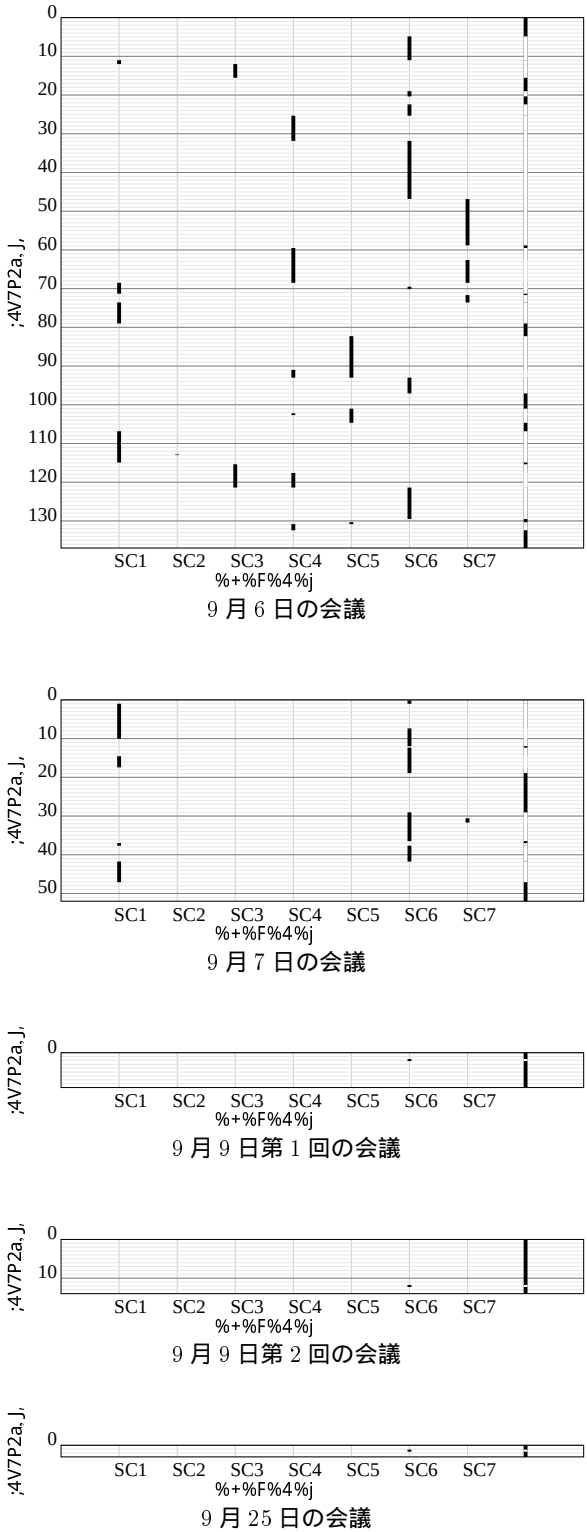
ある SC に属する CI の参照頻度の段階的変化を, 図 5にグラフ表示する. ただし, 1 つの CI のみを持つ SC2 は, 変化を表示することができないのでグラフは除外してある. 例えば SC1 は参照頻度のピークが 3 箇所存在するが (表 3参照), その様子は図 5の一番下のグラフに示されている.

我々の分析対象にした設計者会議は, それぞれの SC に対応する CI の参照頻度が SC1, SC2, SC3 では 0.5 以下の値を示した. それに対し, SC4, SC5, SC6, SC7 は 0,5 を越えた値を持っており, 対応している仕様書の部分は, 設計過程の記録からみて高い品質を持つと予想される.

図 6に, この事例の最終仕様書におけるそれぞれ SC に関する文書の割合を縦軸に示し, 横軸に, それぞれの SC の参照頻度を示す. この図から, 参照頻度と文書の割合には特に相関はないことがわかる. すなわち, 多くの分量を使った部分だからといって, 十分に議論されるわけではない. よって, 参照頻度は文書の量だけからは得られない指標であると言える.

表 5: SC3 の参照頻度

No.	date	提案	言明	確認	参照頻度
1	9 月 6 日	4	0	0	0.00
2	9 月 6 日	0	0	3	1.00
total		4	0	3	0.43



9 月 6 日の会議

9 月 7 日の会議

9 月 9 日第 1 回の会議

9 月 9 日第 2 回の会議

9 月 25 日の会議

図 4: 各会議における CI の分布

表 6: SC4 の参照頻度

No.	date	提案	言明	確認	参照頻度
1	9 月 6 日	1	1	0	0.50
2	9 月 6 日	0	1	1	1.00
3	9 月 6 日	1	0	0	0.00
4	9 月 6 日	1	0	0	0.00
5	9 月 6 日	1	1	1	0.67
6	9 月 6 日	0	0	1	1.00
total		4	3	3	0.60

表 7: SC5 の参照頻度

No.	date	提案	言明	確認	参照頻度
1	9 月 6 日	3	1	0	0.25
2	9 月 6 日	0	0	3	1.00
3	9 月 6 日	1	0	0	0.00
total		4	1	3	0.50

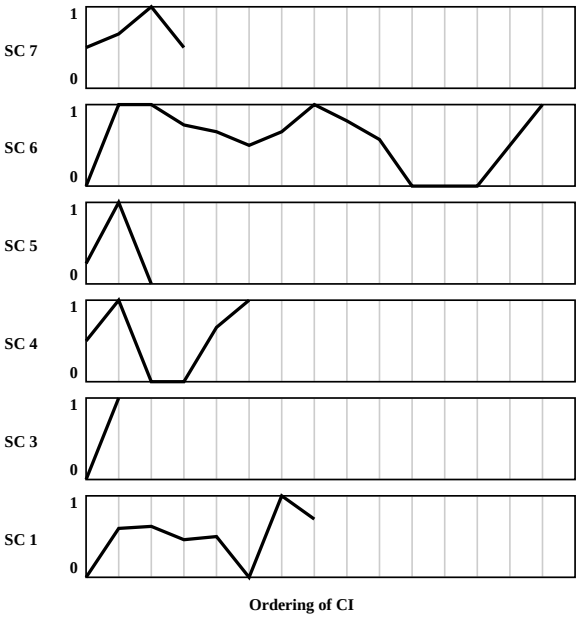


図 5: 参照頻度の段階的変化

表 8: SC6 の参照頻度

No.	date	提案	言明	確認	参照頻度
1	9 月 6 日	9	0	0	0.00
2	9 月 6 日	0	0	1	1.00
3	9 月 6 日	0	2	0	1.00
4	9 月 6 日	2	2	4	0.75
5	9 月 6 日	1	0	2	0.67
6	9 月 6 日	1	1	0	0.50
7	9 月 6 日	1	2	0	0.67
8	9 月 7 日	0	1	2	1.00
9	9 月 7 日	1	0	4	0.80
10	9 月 7 日	3	2	2	0.57
11	9 月 7 日	2	0	0	0.00
12	9 月 7 日	3	0	0	0.00
13	9 月 9 日第 1 回	1	0	0	0.00
14	9 月 9 日第 2 回	1	0	1	0.50
15	9 月 25 日	0	1	1	1.00
total		25	11	17	0.53

表 9: SC7 の参照頻度

No.	date	提案	言明	確認	参照頻度
1	9 月 6 日	2	2	0	0.50
2	9 月 6 日	1	1	1	0.67
3	9 月 6 日	0	0	1	1.00
4	9 月 7 日	2	0	2	0.50
total		5	3	4	0.58

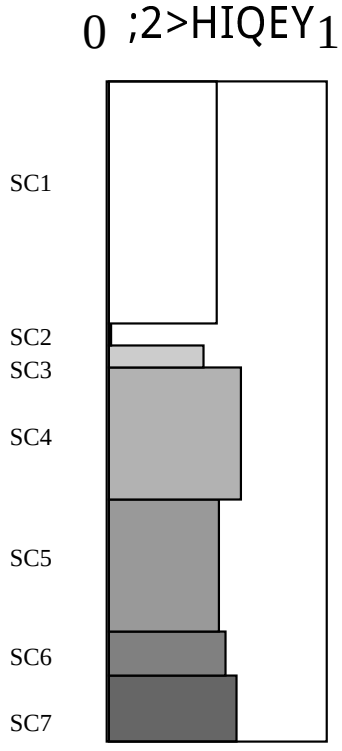


図 6: 最終仕様書における SC の割合と参照頻度

4 議論

この章では, 我々の分析方法の利点/欠点について議論を行なう. 図 4に示すように, 我々が仕様書から選んだカテゴリ SC では分類できなかった会話の時間帯が 31%存在する. これらの中には, もちろん, 会議に関係ない雑談もいくらか含まれているが, その大半は, このシステムのプラットフォームとなる環境についての考察や, 自分たちの作業進行の取り決めや, 2 人の設計者間の作業分担や, 全体会議におけるプレゼンテーションの打ち合せなど, 仕様書から得られた生産物に関する議題ではなく, 作業過程に関する議題が議論されている. これらの作業過程に関する議題の会話は, 最終仕様書からカテゴリを決定する我々の方法では分類仕切れない. また, システムのプラットフォームに関する記述は仕様書から得られても良いはずだが得られていないのは, これに関する話題が最終仕様書に残っていなかったからである. これに関しては, 議事録や会議参加者のメモ等といった会議の生産物からも, カテゴリを設定できれば, 解決できると思われる.

我々の方法では, 仕様書のある特定の部分に結びつけられた会話の断片の列の順序情報に注目する. これは, 会話の時間の長さよりも, 順序の方が有益な情報を含んでいると考えるからである. この理由の 1 つは, 会話の内容がどれだけ生成物に貢献しているかを示す会話の密度の情報を得るのが困難だからである. 単純に短時間に高い参照頻度を出したから, その部分は高い質を持ち, そうでない部分は質が低いとは言い切れない. しかし, 作業過程において繰り返し議論が行なわれ, 参照頻度が向上していく仕様書の部分は信頼のおける部分であると考えられる.

5 おわりに

我々の今回の報告では, SC 間の相互の関係について言及せず, それぞれの SC 内だけの情報で, その評価を行なっている. しかし, CI 間の時間的な重複や, 隣接を基に, それに対応する SC 間の関係を明らかにすることが可能であると考えられる.

我々の分析に用いた原データであるビデオカメラによる記録から, 計算機である程度処理可能なデータを作成するのは, 現段階では人手に頼っている. 今後, 制御のための制御領域を持つビデオテープ再生機を用いて, 会話記録のデータを整理するツールの構築が必要である. このようなツールは, 既存の設計法や言語を用いた設計作業の分析にも容易に適用でき, それらを評価/分析するための方法と合わせることで, ソフトウェア開発プロセスに関する実験的研究に貢献するものと考えられる. 例えば, 実際の作業が既存の枠組で記述できるかどうかを調べる研究 [12] などの研究への応用が期待できる. 本報告の位置付けは, 設計会議における会話を評価/分析するための一指標を与えたことにある.

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] G. L. Rein and C. A. Ellis. rIBIS: a real-time group hypertext system. *International Journal of Man-Machine Studies*, Vol. 34, No. 3, pp. 349–368, Feb 1991.

[3] C. Potts and G. Bruns. Recording the Reasons for Design Decisions. In *10th International Conference on Software Engineering*, pp. 418–427, 1988.

[4] J. Lee. Extending the Potts and Bruns Model for Recording Design Rationale. In *13th International Conference on Software Engineering*, pp. 114–125, 1991.

[5] 岸本三江. ソフトウェア生産プロセスにおけるインターラクションの分析. 情報処理学会第 35 回全国大会論文集, pp. 1141–1142, 1987 年後期.

[6] Bill Curtis. Implication from empirical studies of the software design process. In *Proceedings of an International Conference organized by the IPSJ to communicate the 30th Anniversary*, 1990.

[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.

[8] 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モデル化. 情報処理学会ヒューマンインターフェイス研究会, Jul 1991.

[9] 本間学, 海谷治彦, 佐伯元司. ソフトウェアの実現段階における会議の分析. 情報処理学会 第 44 回全国大会論文集, Mar 1992.

[10] 石井裕. グループウェア技術の研究動向. 情報処理, Vol. 30, No. 12, pp. 1502–1508, Dec 1990.

[11] 西, 海谷, 佐伯. ソフトウェアの発注者-開発者会議におけるインタラクションの分析. 情報処理学会ヒューマンインターフェイス研究会, Mar 1992.

[12] 吉府研治, 垂水浩幸, 西田幸雄. 設計理由記録モデルの比較. 情報処理学会第 44 回全国大会論文集, pp. 5–351, Apr 1992.