仕様作成作業における
認識の不一致を検出するための基礎実験

海谷治彦

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

共同作業を行う作業者にとっては，作業に関する相互理解が重要となってくる．しかし，相互理解が図られていないことを，作業者自身が気がつかない場合もあり，それが理由で作業の効率が低下したり，矛盾を含む生産物が作成されたりする．本稿では，実際に共同作業を行っている作業者の活動から観測可能な事象をもとに，作業に関する認識が一致していない兆候を発見するための基礎実験の報告を行う．本実験では，特に，二人の作業者がソフトウェアの設計を図表現で行う作業に対象を絞る．それぞれの作業者には異なる図表現を利用させるため，異なった視点からの設計が行われる．この作業は以下の点で認識の不一致に関する研究や実験を行う場合に適している．

- 視点が異なるために，単純な referential task ではないこと．
- 作成対象が図であるために，観測可能な事象が豊富なこと．

最後に，このような作業における認識の不一致の検出を，計算機によって支援する可能性と限界について議論する．

Preliminary Experiments for Detecting Misunderstanding in Software Design

Haruhiko Kaiya

Graduate School of Information Science,
Japan Advanced Institute of Science and Technology, HOKURIKU

Tatsunokuchi, Ishikawa, 923-12 Japan

Email kaiya@jaist.ac.jp
URL http://www.jaist.ac.jp/~kaiya/

It is necessary for the cooperative workers to have mutual understanding with each other. But they sometimes can not realize that they do not have, hence the efficiency of the task goes down and inconsistent products are produced. In this paper, I report preliminary experiments for finding symptoms of misunderstandings among the cooperative workers from the observational phenomena in their activities. In the experiments, I focused on a task, namely *negotiation task for software diagrams*, where the design for a software was represended in a form of diagrams. This task is good both for studying misunderstandings and for having experiments, because it is not a kind of simple referential task, and a lot of observational phenomena, such as drawing, pointing and talking during their activities, can be observed in the task. From the results of the experiments, I discuss the possiblities and limits of the computer support for detecting misunderstandings.

1　はじめに

共同作業を通して，なんらかの生産物を作成する場合，他者との合意形成をもとにした意志決定，場合によっては折衝や妥協なども必要となる．しかし，合意形成や意志決定，折衝や妥協が正しく行なわれるためには，共に作業する他者との間に，作業対象のことがらについての認識が一致していることが前提となる．すなわち，話題となっている事柄を，他者はどのように理解しているか，もしくは，話題としている事柄自身が作業者間で共有されているかなどが保証されていなければ，意味のある合意，意志決定，折衝などは期待できない．特に，ソフトウェア開発などに代表される決定の積み重ねを必要とする一連の作業では，初期の段階の問題点が，後段階の作業に対して致命的な打撃を与えることは良く知られている [1]．しかし，互いの認識が一致していないことを作業者自身が認識していないために，非効率的な作業や，互いに整合性のとれない生産物群などが発生する場合がある．

よって，共同作業の支援の一つとして，共通理解の不一致を検出する手法やツールが必要となる．その一例として，KJ 法などの図解法をもとにして，共通理解の一致/不一致を表示する方法などが提案されている [2]．また，設計作業履歴 [3] を，作業の参加者や，参加できなかった関係者が参照することで，作業に関する認識の不一致を軽減することが可能であると思われる．しかし，これらの技術を適用するために従来の作業手順を変更することは，実際には困難な場合が多い．そこで，従来の作業方法をできる限り阻害しない方法で，認識の不一致の発生を検出する方法が望ましいと思われる．

本稿では，作業中に外部から観測可能な活動をもとに，認識の不一致の発生を検出するため基礎実験の報告を行う．実験対象は，ソフトウェアの設計図を分担して作成することに限定する．続く 2節では，この活動を調査するために設計した作業形態について説明する．さらに，3節と 4節では，この作業を用いた実験と分析の手順を示す．そして，5節で実験の結果と考察を行い，最後に，認識の不一致の検出の計算機システムによる支援の可能性と，その限界について議論する．

2　ソフトウェア設計図の擦合せ作業

ソフトウェア開発においては，異る視点から設計対象を扱う設計法を組み合わせて利用する場合が多い．メソッドエンジニアリングと呼ばれる設計方法論の研究 [4] では，適材適所に設計法を組合わせて利用する場合も見られる．このような場合，それぞれの設計法を複数の作業者で手分けして利用することは容易に想像できる．例えば，ソフトウェアの分析/設計法である OMT[5] には，機能モデルと呼ばれる方法と動作モデルと呼ばれる方法が含まれている．そこで，これら二つを二人で手分けして行うことが可能である．また，OMT の例のように，設計法の多くは図表現を用いた言語を利用する．当然，手分けした設計図を最終的に 1 つの対象の設計とするために，作業者間で擦合せの作業が必要となり，必要であれば設計を修正/追加する必要が生じる．このような作業を本稿では，*NTSD* (Negotion Task for Software Diagrams) と呼ぶことにする．NTSD を円滑に行うことは，矛盾のない開発を行うために重要な要件となる．よって，本研究では，上記のような作業における認識の不一致の検出方法について議論する．

本作業は，以下に示すような点から，認識の不一致に関する研究を実験的に行なう場合に適している．

- 作業が作図であるため，分析者が観測可能で，かつ計算機で自動的に測定可能な事象を多く含む．
- 作成する図の構文が明確なため，作業の分析が容易である．
- 同じ対象を異なる観点から取り組む要素が強いため，Referential Task に関する研究成果 [6], [7] を単純には適用できない．

3　実験の目的と手順

実験の目的は，NTSD における認識の不一致を反映する事象を調査することにある．そのために実験と分析は以下の要件を満たさなければならない．

- 認識の不一致が発生している個所を特定できること．
- 被験者は，できる限り，自然な形で NTSD を遂行できること．
- 不一致を反映すると思われる事象，発話や描画行為などを，作業自体を阻害せずに記録できること．

本節と 4節で実験の手順と分析の手順について説明する．説明では，後述の 5節で紹介する酒問屋の業務の分析 [8] を，OMT 法の機能モデルと動作モデルで設計する事例を利用する．図 1に手順全体の概要を示す．

実験者は，以下の手順で被験者に対して指示を行ない実験を進める．

1. それぞれの被験者に自然言語で書かれた要件と簡単な構造モデルを渡し，機能モデルもしくは，動作モデルでの業務分析を行なうように指示する．機能モデルではデータフロー図 (DFD) を作図する作業が中心となり，動作モデルでは状態遷移図 (STD) を作図する作業が中心となる．要件は，20 文程度の曖昧な部分を多く含むものである．さらに，後日，自分の担当でない方の図を記述した被験者 (パートナー) との摺合せ作業 (会議) を行なうことを告知する．また，会議日までは，パートナーとは連絡をとらないように指示する．

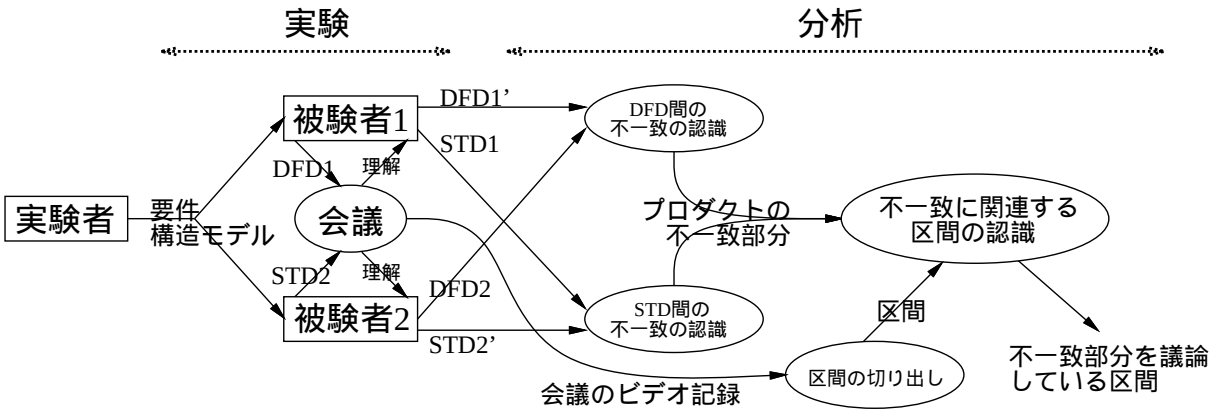

図 1: 実験と分析の手順

2. 2 つのモデルの摺合せ作業 (会議) を行なってもらう . 被験者は , 自分の書いてきた図 , 要件 , 教科書などを参照しながら議論してよいが , それぞれが準備してきた図を直接相手に見せることはせず , 黒板に写しながら説明を行なう . また , 摺合せの基準として , 自分の書いた図の修正が最小となるように指示する .
3. 摺合せ後に , 被験者に , 会議の議論を踏まえて , それぞれ個別に機能モデルと動作モデルの両方を記述するように指示する . すなわち , 機能モデルを担当した被験者 1 に , 自分が前もって書いた DFD の改定版である DFD1' と , 会議での理解をもとに STD1 を記述してもらう . 被験者 2 も同様である . この際 , 会議中のメモ , および他者の準備してきた図は実験者の方で回収し , 被験者には参照させない .

上記手順 3 における摺り合せ会議後の作図は , 記憶の劣化の影響を最小にするため , 会議終了後 , できるだけ早く , それぞれの被験者に記述させた . また , 会議中に , 他者の図を記憶することに専念させないために , 他者の分担の図を会議後に記述する指示は会議の後に行なう .

実世界の作業では , 摺合せ後に他者の分担部分まで記述することはほとんどなく , 必要ならば自分の分担部分を修正する程度だと思われる . また , ソフトウェア工学的な観点からも上記の作業方法は現実的/理想的ではない . しかし , 本稿で提案する手法の目的が , 摺合せ会議を行なった後でも , なお内在する共同作業者間の認識の不一致を作業過程の情報から獲得しようとするものである . よって , あえて本実験では摺合せ後に他者の分担の図まで記述させることにした .

4 分析方法

分析では , それぞれの被験者が会議後に作成した図の差異をもとに認識の不一致の存在を確認し , 不一致部分に関する会議中の区間の性質を調べることが目的となる . 本稿では , 特に会議中の作業者の描画に関係する行為に注目して分析を行う . 描画に関わる作業は , 発話や会話と比べ , 現在利用可能な計算機でも容易に検出が可能であるからである .

以下に図 1に従いその手順を説明する .

1. 会議内容のビデオカメラによる記録を , 作業者の描画などの行為を基に , 内容がまとまった区間に分ける .
   特に , 会議中には , 議論対象となる図や資料は , 何回も切り替わるため , このような切り替わりに注目する . 例えば , 会議などでは , 複数の図を書いたり , 資料や参考書を途中で参照することで , 議論の中心となる資料が刻々と変化する .
2. 分析者は被験者 1 の DFD1' と被験者 2 の DFD2 , 被験者 1 の STD1 と被験者 2 の STD2' の差異をもとに認識の不一致を認識する .
3. 認識の不一致に関する議論が行なわれている区間の活動の性質を調べる . 特に , 区間内での描画に関わる参加者の操作に注目する .

認識の不一致は会議後に作成された図の差異をもとに決定する . まず , 図中の節や枝の数 , そこについた注釈 , 節や枝の間の接続の違いを構文不一致とする . 注釈の違いのうち , 内容的に同一のものは差異とはしない . 例えば , "数" と "数量" などの差異 , 複合データ構造を , そのまま注釈として記述してあるか個々のデータに展開してあるかなどは差異とはしない . さらに , いくつかの構文不一致がまとまって内容的な不一致を構成しているため , それを内容不一致とする . 内容不一致の中でも接続の違いを含むものを重大な不一致とする .

図 2に , 構文不一致と内容不一致の例を示す . 図中の 2 つのデータフロー図では , アクタからのデータの流れ先と , プロセスとデータストアの流れの方向が逆であるため , 構文不一致は 2 であるが , あるデータフローをどのタイミングで保存するか , 二次加工するかの違いと捉えると内容的には 1 つの不一致と考えられる場合が

ある．また図のトポロジが異なるため重大な不一致でもある．

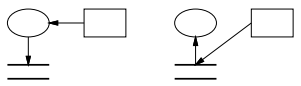

図 2: 構文不一致と内容不一致

5 結果と考察

5.1 区間の分割

被験者は本学の OMT の講義の合格者から募った．表 1に被験者と作業の概要を示す．被験者が注目している図の切り替わりを用いて分割した区間の平均時間は，面識がある場合は若干長いが，それほど大きな差はない．また，平均区間長は，区間内の議論内容を概観するのに十分小さい長さであると思われる．

表 1: 被験者ペアの概要と作業時間

	面識	会議時間 (分)	平均区間長 (分)
ペア 1	なし	140	1.7
ペア 2	あり	50	2.1

5.2 不一致の識別

表 2, 3に，会議後にそれぞれに作成された生産物 (図) の規模と，それらの図の差異から見た認識の不一致の程度を示す．規模は節の数と枝の数のペアで示し，重大な不一致の数はカッコ内に付記している．また，それぞれの表中の被験者 1，被験者 2，DFD1', DFD2, STD1, STD2' などは，図 1内のそれと対応している．比較する図の間において，図の規模の差異はほとんどないが，かなりの数の構文的な差異が見られる．その多くは節や枝への注釈の違いなどであった．しかし，双方のペアにおいて，トポロジの違いを中心とした重大な認識の不一致が見られた．

5.3 不一致区間の特徴

本分析では，以下に示す図への操作に注目した．
point : 図中の特定の要素や部分を指などで指す
trace : 図中の複数の要素間の関係を指などでなぞる
draw : 図に新しい要素を書き加える
delete : 図の要素を削除する

以下に，それぞれのペアの代表的な不一致についての分析を例示する．図 3と 5には，不一致が発生していた部分 (DFD1' と DFD2) と，会議中に描画されたその部分の中間生成図 (DFD1.1 から 1.4 など) を示す．DFD1' と DFD2 は図 1中のそれぞれに対応している．また，図

表 2: 図の規模と認識の不一致の概要 (ペア 1)

	機能モデル	動作モデル
被験者 1	13 + 16 (DFD1')	7 + 10 (STD1)
被験者 2	13 + 16 (DFD2)	6 + 12 (STD2')
構文不一致数	16	6
内容不一致数	6 (2)	4 (0)

表 3: 図の規模と認識の不一致の概要 (ペア 2)

	機能モデル	動作モデル
被験者 1	15+20 (DFD1')	7+10 (STD1)
被験者 2	12+15 (DFD2)	7+10 (STD2')
構文不一致数	20	8
内容不一致数	6 (3)	3 (1)

4と 6は，それぞれの中間生成図において行われた操作の時系列を，図中のどの部分に関するものかと，どちらの被験者によって行われたかを付記して示している．また，その区間が会議中のどの時間帯で行われたかも付記してある．例えば図 3の DFD1.1 の中間生成図は，会議開始から 64 分くらいから 66 分くらいの 2 分間に議論対象となったことが図 4に示してある．

5.3.1 ペア 1(面識なし)

図 3での不一致は，

- データストア C は，A と B のどちらで作成されるか?

ということが中心となっている．

図 4の中間生成図 DFD1.1 と 1.2 を操作する区間では，被験者 2 が主に DFD2 に該当する部分を記述している．この DFD2 は被験者 2 が会議後の最終生産物として提示した図の一部である．それに対し，被験者 1 は，その記述に対して同意しているようである．それは，被験者 1 が，その記述を trace する操作を何回か行っていることからわかる．

しかし，図 4の中間生成図 DFD1.3 で，被験者 1 は，DFD1' に該当する図に，また戻してしまう．この DFD1' は被験者 1 が会議後の最終生産物として提示した図の一部であり，本来の DFD の分担者である被験者 1 が事前に準備した図とほぼ同じである．また，この時区間では，被験者 2 による操作がほとんど行われておらず，行われた場合も，ここでの被験者 1 の修正操作 (draw と delete) に沿わない操作 (部分 1 8 2 9 4 8 を point する) を行っている．

上記の事例から，会議のより最後において，修正操作 (draw や delete) が，修正を行った作業者以外に追認されない場合，認識の不一致の発生の危険が高いと思われる．また，分担して作業を行っている場合，事前に準備してきた部分の修正や変更が行われる場合も同様に不一致が発生しやすいと思われる．

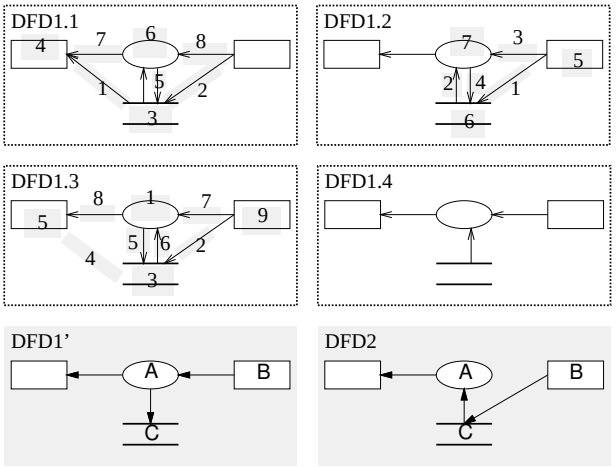

図 3: 中間生成図と最終図 (ペア 1)

DFD1.1 (64-66)

area	who	how(action)
1	subject1	point
1 2	subject2	draw
2 3 1 4	subject2	trace
2 3 5 6 7 4	subject1	trace
1	subject2	delete
6 3	subject2	point
3 5 6	subject2	trace
5	subject2	draw
2 3 5 6	subject2	trace
6	subject2	point
6	subject1	point
3 5 6 7	subject1	trace
8	subject1	point
2 3 5 6 7	subject1	trace

DFD1.2 (70-71)

area	who	how(action)
1 2	subject1	trace
3	subject1	point
3 4	subject2	delete
5 1 6	subject2	utter
7	subject2	point
7	subject1	utter

DFD1.3 (72-78)

area	who	how(action)
1	subject2	point
2	subject1	delete
3 2 3 4	subject1	point
5	subject1	draw
6	subject1	trace
7	subject1	draw
7 1 3	subject1	trace
1 4	subject1	trace
1	subject1	point
1 5 3	subject1	trace
1 4	subject1	trace
1 8 2 9 4 8	subject2	point
5	subject1	delete

図 4: 図 3での操作列

5.3.2 ペア 2(面識あり)

図 5での不一致は，

- プロセス A からの出力データフローは，B と C のどちらに流すべきか?
- また，データ b はどこで生成されるのか?

ということが中心となっている．特に前者の部分は会議開始前の図にはない部分であるため，ペア 1 の例とは異なり，事前に準備された図の影響は少ないと思われる．

図 6の中間生成物 DFD1.4 に関する操作の区間以外では，それぞれの区間において，どちらか一方の被験者のみが操作を行っている．DFD1.1 の区間では，被験者 1 が自分の事前に準備してきた部分 (DFD1.1 中の 1, 4, 2 など) を point しているが，それに加えて 4 と 2 の流れを統合する別解 (DFD1.1 中の 3) についても trace によって示している．

DFD1.2 に関する区間では，被験者 2 によって，DFD1.1 でも現れた別解 (3 の trace) が，同様に trace されている．これは，この別解が被験者 2 にも支持されたことを表していると思われる．さらに，この区間では，被験者 2 は，1 の部分を trace することで，新しい話題を提案している．DFD1.3 の区間では，この新しい話題に対応する部分を被験者 1 が記述を追加している．

DFD1.4 に関する区間においては，はじめに，DFD1.2 中の部分 1 に対応するが接続が異なる部分 1 が被験者 1 によって記述される．それに対し，被験者 2 は，被験者 1 がこの区間で追加した部分 1，DFD1.2 中で自身が提案した部分 1 に対応する部分 2 の順で trace を行っている．これは，被験者 2 が後者の方法 (部分 2) も考慮に入れるべきだという意思表示だと考えられる．しかし，被験者 1 はこの後に前者の方法 (部分 1) の流れを trace することで，被験者 2 の提案を拒否しているようである．この拒否に対する被験者 2 の合意の意思表示は，point などの操作，および発話内容を聞いてさえも確認することはできなかった．

このペアの例でも，決定を伴う操作への他者の追認が行われないことが，認識の不一致地点を示唆する 1 つの指標となっている．

6 おわりに

図に対する操作と操作者の認識は現在の計算機でも可能であると思われるが，視線や発話によるポイント，他者の指示による図に対する操作などがある場合，本方法の適用は困難となる．この場合，支援システムのオペレータなどが，これらの情報を計算機に対して与える必要がある．また，本手法では，話者交替の情報は利用していないが，その情報を用いて，より明確な検出をすることが可能であると思われる．話者交替の情報を利用

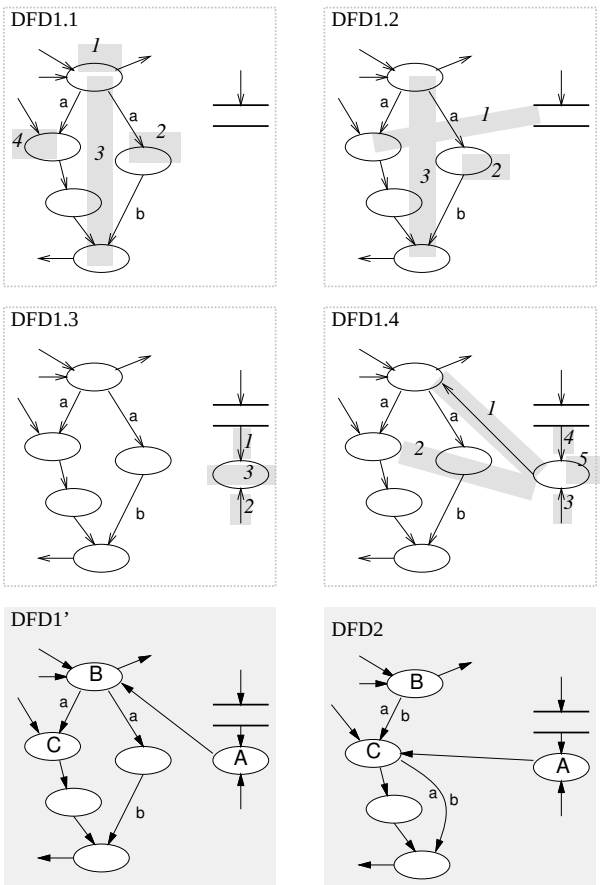

図 5: 中間生成図と最終図 (ペア 2)

DFD1.1(26-30)

area	who	how(action)
1 2	subject1	point
3	subject1	trace
4	subject1	point

DFD1.2 (37-38)

area	who	how(action)
1	subject2	trace
2	subject2	point
3	subject2	trace

DFD1.3 (39-40)

area	who	how(action)
1 2 3	subject1	draw

DFD1.4 (42-48)

area	who	how(action)
1	subject1	draw
1	subject2	trace
2	subject2	trace
3 4 5 1	subject1	trace

図 6: 図 5での操作列

した話題の認識は文献 [9] におけるシステムに実装されている．さらに，単純に認識できる範囲の操作や発話の言語行為情報を用いれば，より強力な検出を期待できるが，それらの認識を計算機で行なうことは困難であるため，前述の視線などの情報と同様に人間による補間作業が必要となる．

OMT のような作成要素 (データフロー, データストアなど) の認識が容易な作業では，本稿の手法は容易に適用できるが，構成要素の認識が難しい議論では本方法の適用が困難となる．しかし，最終的に図や文書などの生産物を作成することを目的とする共同作業の場合，その生産物の構造は，ある程度予測ができるため，そのような目的の作業に適用する限りにおいて，問題にならないと思われる．

謝辞

本研究の一部は，文部省科学研究費 奨励 (A) 課題番号 08780257 の援助の下に実施された．

参 考 文 献

[1] B. Boehm. Industrial software metrics top ten list. *IEEE Software*, Sep. 1987.

[2] K. Hori. A System for Aiding Creative Concept Formation. *IEEE Trans. Systems, Man, and Cybernetics*, Vol. 24, No. 6, pp. 882–894, 1994.

[3] 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.

[4] S. Brinkkemper. Method engineering. *Information and Software Technology*, Vol. 38, pp. 275–280, 1996.

[5] James Rumbaugh, et al. *Object-Oriented Modeling and Design*. Prentice-Hall, 1991.

[6] H. Clark, et al. Refering as a collaborative process. *Cognition*, Vol. 22, No. 1, pp. 1–39, Feb. 1986.

[7] R. M. Krauss, et al. Mutual Knowledge and Communicative Effectiveness. In *Intellectual Teamwork: Social and technological Foundations of Cooperative Work*, chapter 5, pp. 111–145. Lawrence Erlbaum Associates, 1990.

[8] 山崎利治. 共通問題によるプログラム設計技法解説. 情報処理, Vol. 25, No. 9, p. 934, Sep. 1984.

[9] Haruhiko Kaiya, et al. Design of a Hyper Media Tool to support Requirements Elicitation Meetings. CASE'95 Proceedings.