Vol. 40 No. 11 情報処理学会論文誌 Nov. 1999

代案発生に注目した ソフトウェアの共同設計作業における認識不一致の分析

海 谷 治 彦†

設計作業を分担して行う場合，設計者間での認識不一致を早期に発見し解消することが重要である．分担部分間の関係を明確に定義することで，この問題を解決するアプローチもある．しかし，関係を明確に定義できない場合には不一致が発生していること自体の認識でさえ困難である．そこで本稿では，分担部分間の関係を直接に調べることが困難な場合における認識不一致の特徴を調べるための実験・分析法を提案し，その適用例を示す．本方法では，ある作業部分における担当者と非担当者の理解の差を認識不一致と定義し，その不一致の発生を反映している共同作業の特徴の抽出を行う．OMTを用いたソフトウェア設計作業に本方法を適用した結果，合意のうえで却下された部分は不一致が少ないこと，不一致となる部分は代案を持つ場合が若干多いこと，調整作業中における代案どうしの発生パターンに特徴があることが分かった．

An Analysis of Misunderstandings during Cooperative Software Design with Respect to the Occurrences of Alternatives

Haruhiko Kaiya†

It is difficult for collaborative designers to successfully complete their product if they misunderstood with each other. Furthermore, if the parts of the product have no explicit relationships for managing their consistencies, they can not find and correct their misunderstandings by comparing their products directly. In this paper, we propose an experimental method for investigating the characteristics of design processes that are related to the occurrences of misunderstandings. And we apply this method to several pairs of software designers. In this method, we define the misunderstandings as the differences between a product revised by a designer taking its charge, and its image supposed by the other designers. From the result of our experiment, we found that few misunderstandings are found in mutually reject parts, and that the rate of misunderstood parts which have their alternatives are slightly larger than the rate of mutually understood parts, and that the occurrences of alternatives during the coordination phase have a specific characteristics if the alternatives are mutually understood.

1. はじめに

製品や文書などを複数の作業者で作成しなければならない場合，互いの認識が一致していることが重要である．しかし，作業者間の認識が一致しておらず，なおかつ不一致が発生していること自体を作業者が認識していない場合，円滑な共同作業を行うことは困難であると思われる．特に，作業が時間的に長期間にわたる場合，初期に発生する認識の不一致は，後に続く作業や生産物に重大な悪影響を及ぼす．そこで，作業が終了してからではなく，作業中において早期に認識の不一致を発見する必要がある．

† 信州大学工学部情報工学科
Department of Information Engineering, Faculty of Engineering, Shinshu University

ソフトウェア開発の分野においは，作業者が分担して作成する部分間の関係を明らかにすることで，上記のような問題の解決に取り組んでいる．たとえば，文献 1) では，異なる設計モデルを形式的に統合する方法が提案されている．具体的には，代表的なオブジェクト志向設計法である Object Modeling Technique (以降 OMT)[2] において記述される複数のモデルを数学的に単一のモデルに統一することで，モデル間の無矛盾性を維持することを可能としている．また文献 3) では，ViewPoint と呼ばれる個々のモデルの構文，記述プロセスなどを整理するためのデータ構造が定義されている．その中の 'Inter-ViewPoint Check Action' と呼ばれる項目に，モデル間の関係を明示的に記述することで，不一致や矛盾の発生を検出し，それを解消することを支援している．

しかし，このように異なるモデル間の関係を明示的に記述するアプローチでは扱いきれない場合もある．たとえば，近年，イベント駆動型システムに代表されるような独立部品型のアーキテクチャ[4]を持つシステム，たとえばGUIや分散オブジェクトシステムなどの重要性が高まっている．これらは，たとえ作業分担をモジュール単位に行ったとしても，モジュール間の関係を明示的に記述することは容易ではない．

そこで，部品間の関係に明示的に言及することなく，すなわち，認識を一致させるべき対象を直接に比較することなしに，認識が一致しているか否かを判定する手法が必要となる．認識の一致/不一致を反映している現象として有力なものは，作業者間のインタラクションである．作業者間のインタラクションの代表例として会話があり，言語学の分野において共通理解が成就される場合の会話の特徴に関する理論がいくつか提案されている[5)~8)]．よって，これらを直接利用することで認識不一致の発生を検出することができるように思われる．しかし，これらの特徴を得るためには，通常，会話の内容に深く踏み込まなければならず，その適用コストは莫大なものとなる．また，より詳しい内容の分析をすることで，認識の不一致個所の検出が高精度になる保証はない．

設計作業の履歴を構造的に記録し，設計過程での推論過程を支援したり，作業に関与する作業者間のコミュニケーションを支援したりするために，Design Rationale (DR) 技術[9]が提案されている．ほとんどのDR記述は，解決すべき問題点に対して，複数の代案をおき，それをある基準で評価し比較する構造となっている．このような構造を明確にすれば，認識不一致などの発生を抑制することも期待できる．しかし，DR記述を厳格に行うことはかなりのコストがかかるうえ，人間の自然な思考過程を阻害する恐れもある．さらに，文献10)によると，時間とともに技術，設計者の技量，組織の方針などが変化するため，DR記述の寿命は短命であることも指摘されており，個々のDR記述に十分な手間をかけることは非効率的であると思われる．また，記録コストや寿命を無視したとしても，意識的に記述できる理由には限度があり，意識的に記述できない理由や‘思い違い’が重大な認識不一致を引き起こすと思われる．文献10)の調査によると，DR記述を用いても，設計理由に関する疑問の41%しか答えられなかったそうである．

そこで本稿では，ある担当部分に関する担当者と非担当者の理解の差を認識の不一致とし，その発生の特徴を分析する．これによって，異なる分担部分間の関係に頼らずに認識の不一致を判定できる．また，作業過程における個々の作業の意味には踏み込まず，どの部分について作業，もしくは議論しているかについてのみに注目する．これによって黒板型計算機[11]などを用いた不一致部分検出の自動化が期待できる．この方法をOMTを用いた設計作業に適用し，実際にどのような作業過程の特徴が認識不一致/一致を反映しているかの調査結果を示す．

本稿の構成は以下のとおりである．まず2章では，本稿で対象とする共同設計作業をモデル化し，その上で認識不一致の定義を行う．認識不一致は実作業では観測できない生産物をもとに定義されている．そこで，認識不一致を間接的に反映していると思われる作業過程上の特徴について議論する．3章では，認識不一致の近似値を求め，その近似値が作業過程の特徴とどのような関係があるかを調査する実験・分析法を具体例を用いて説明する．4章で実験の適用結果の例を示し，5章で本稿をまとめる．

2. 共同設計作業モデルと認識不一致

本章では，共同設計作業を，その生産物，担当者，作成時期などをもとにモデル化し，そのモデルをもとに認識不一致を定義する．さらに，認識不一致の発生に影響を与えていると思われる要因について述べる．

2.1 共同設計作業のモデル化

前述のとおり，本稿では分担部分間の直接的な比較が難しい場合の認識不一致を取り扱う．このような不一致をモデル化する基盤として，Clarkの談話の理論[12]を用いる．この理論では，共通理解は以下の手順で成就されるとしている．

(1) 説明者がある情報を伝達する．
(2) 聞き手がそれを理解し，どう理解したかを説明者に伝達する．
(3) 聞き手の伝達を説明者が本当に理解できているか否かという観点から評価する．
(4) その評価を説明者が聞き手に伝達する．

ここで共通理解を認識の一致と見なし，そうでない場合を不一致とする．たとえば，図**1**に示すような会議では，ある部分の担当者の担当部分の理解と，非担当者のその部分の理解の差分をもとに認識の一致・不一致を検出する．

以下ではモデル化の対象となる作業を定義し，その上で上記のような認識の不一致を形式化する．まず，共同作業を行う作業者はそれぞれに自分が作成義務を負っている生産物を持っており，以下のフェーズを繰り返すことで共同で生産物群を完成させるものとする．

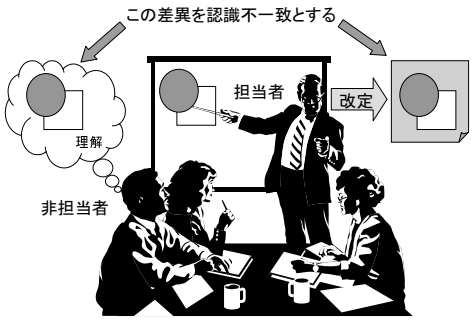


図 1 認識不一致の概要
Fig. 1 Outline of misunderstanding.

調整前フェーズ ： それぞれの作業者が他の作業者と調整をする前に生産物の準備稿を作成する．

調整中フェーズ ： 調整前フェーズで作成した準備稿を用いて，他の作業者との調整作業を行う．調整中に参照されたり，新たに記述されたりする生産物の要素の集合を調整稿と呼ぶことにする．

調整後フェーズ ： 調整中フェーズでの結果をふまえて，必要ならば自分の担当する生産物の改定稿を作成する．

繰り返し調整を行う場合，以前の改定稿が新しい準備稿となる．

それぞれの作業者は，調整後フェーズにおいて改定稿を作成する際，他の担当者がどのように改定されるかを想像したうえで，自分の担当部分の改定を行う．いいかえれば，ある生産物に関しては，その担当者が記述もしくは関与した準備稿，調整稿，改定稿に加えて，実際には記述されないが，非担当者が想定している想像稿が概念的には存在すると思われる．そこで，共同設計作業における認識不一致を，改定稿と想像稿の差分と定義する．以上の草稿の関係は図 **2** のようになる．

草稿中に現れる生産物の要素は，ある 1 つの草稿だけに出現するわけではなく，たとえば準備稿，調整稿，改定稿のすべてに出現する場合もある．そこで，草稿に現れる要素を図 **3** に示す 4 ビットのカテゴリコードで分類する．カテゴリコード 0000 に分類される要素は観測不能なため，15 通りのカテゴリコードを用いる．なお，コード中の値が ? の場合は，0 もしくは 1 のどちらかであることを表す．

2.2 認識不一致と一致

図 3 のカテゴリコードを用いて，認識不一致と一致を以下のように定義する．

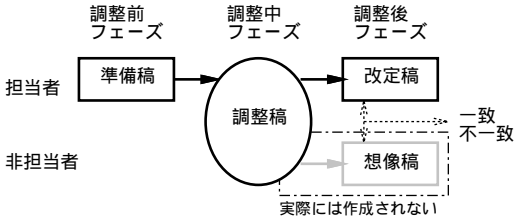


図 2 4 種類の草稿の関係
Fig. 2 Relationships among four kinds of drafts.

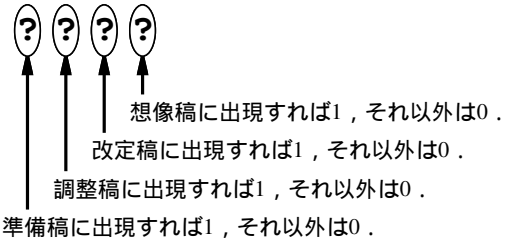


図 3 草稿中の要素を分類するための 4 ビット・カテゴリコード
Fig. 3 Category code for classifying elements among drafts.

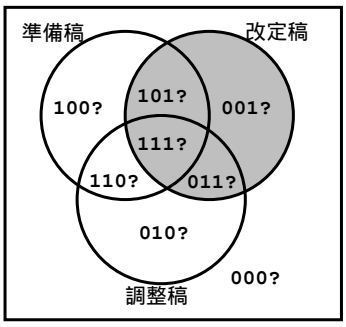


図 4 観測可能な生産物の要素の関係
Fig. 4 A Venn diagram of observational elements of drafts.

陽一致 ： ??11 となる要素．これは担当者および非担当者双方が記述すべきであると認めた部分であり，図 4 の灰色の領域に含まれる．

陰一致 ： ??00 となる要素．これは担当者および非担当者双方が記述すべきではないと考えた部分であり，図 4 の白色の領域に含まれる．

不一致 ： ??01 もしくは ??10 に分類される要素．これは担当者または非担当者のどちらか一方のみが記述すべきであると考えた部分である．

陽一致と陰一致をあわせて単に一致と呼ぶ場合もある．

実際には，‘覚え違い’ に相当するものが ‘認識の不一致’ に混在する可能性がある．2.1 節で述べた Clark の理論では，説明者，聞き手間の複数回のインタラク

ションを通じて共通理解が成立することになっている．そこで，このような複数回のインタラクションが成立していない部分は‘覚え違い’の恐れがあるため分析から除外する．

2.3 不一致を反映する作業過程の特徴

実際には想像稿は作成されないため，カテゴリコードの最後の 1 ビットの情報は利用できない．すなわち，不一致を発見した時点では，図 4 に示す分類までしか行うことができない．しかし，この分類だけからも以下のような仮説を立てることができる．

- カテゴリコード 11?? の領域に比べ，01?? の領域の要素の方が不一致となる場合が多い．なぜなら，準備稿段階で記述されている部分は，担当者が熟考した部分であるため簡単には認識が異なる場合は少ないのに対して，準備中フェーズで発生した部分は，双方ともに十分な検討や理解ができないため，認識不一致が発生してしまう危険が高いからである．
- 10?? の領域には重要な要素が含まれない．なぜなら，準備稿段階では記述があるにもかかわらず調整中フェーズで取り上げないのは，担当者自身が不必要と判断した場合であるため，そもそも検討対象とはならない．

そこで，実際に一致/不一致となる個所と，この分類とに関係があるかを調査する．

次に，草稿中で互いに代案となっている要素間の関係に注目する．代案とは IBIS [13] の Position，QOC [14] の Option に相当する．たとえば，調整フェーズにおいて，準備稿の段階で発案された記述の代わりとなる記述が提案される場合，それらの記述どうしは代案関係にあるとする．代案と不一致の関係に関しては以下のような仮説を立てることができる．

- 代案の存在が不一致の誘因となる．
- 代案の中でもそれが不一致となるか否かは調整中フェーズにおいての代案どうしの発生パターンと関係ある．

そこで，まず代案の存在と不一致の発生との関係を調べ，さらに，調整中フェーズにおける代案どうしの発生パターンを調査する．

3. 認識不一致の発生分析のための実験

2 章で述べたように，実際には想像稿は記述されないため，認識不一致を直接には判定することができない．そこで，認識不一致と，作業過程の特徴の関係を調べる実験を設計した．この実験では想像稿の近似稿を意図的に被験者に作成させることで，認識不一致の近似値を得る．

実験では複数視点からのモデルを記述する必要のある図式言語を用いたソフトウェアの設計作業を対象とし，個々の作業者は異なるモデルを分担することにした．これは，作業者が分担部分間の関係を直接比較することが困難な作業の一例といえる．

3.1 設計言語と対象

OMT によるソフトウェアの設計図を生産物とした．文献 15) の例題をもとに，自然言語で記述した要件 (Requirements Document，以降 RD；30 行程度) と簡単なオブジェクト図 (Object Diagram，以降 OD；10 ノード，11 アーク程度) を実験者が用意し，それぞれの被験者にデータフロー図 (Data Flow Diagram，以降 DFD) もしくは状態図 (State Stansition Diagram，以降 STD) どちらかを作成させる．

3.2 被 験 者

大学院における OMT の授業の合格者，4 ペア (8 人) を被験者とした．うち 2 ペアは互いに面識があり，他の 2 ペアは互いに面識がない．

3.3 実験手続き

図 **5** に実験の流れの概要を示す．

(1) 実験者が被験者それぞれに RD と OD を手渡し，一方 (被験者 1) が DFD，他方 (被験者 2) が STD を作成するように指示する．さらに，数日後にパートナー (他の図を作成する被験者) との打合せ (会議) をするため，その準備をするように指示する．打合せまでは，パートナーとは連絡をとらないように指示する．

(2) 被験者はそれぞれに指示された図 (DFD1，STD2) の準備稿を作成する．

(3) 会議前：調整作業を対面式の会議の形式で行ってもらう．会議室では図 **6** に示すように．大型の黒板 (4.2 m × 1.7 m 程度) とマーカー，テーブルなどが利用可能である．

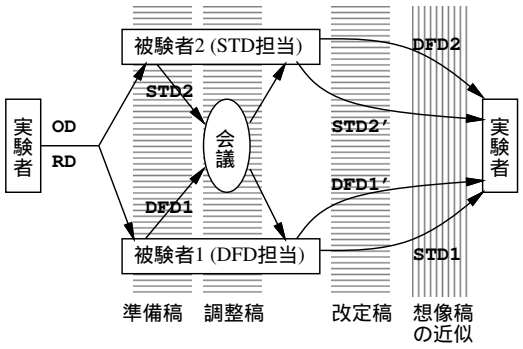


図 **5** 実験の流れ
Fig. 5 The flow of an experiment.

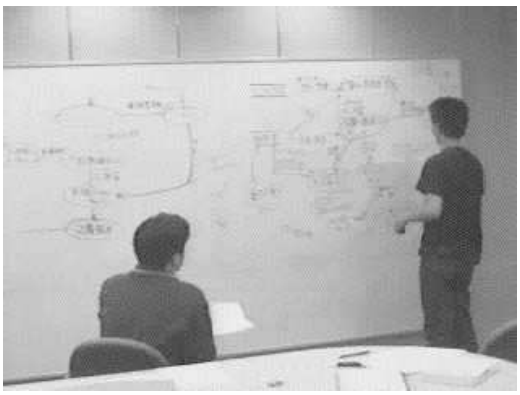


図 6 会議室の設定
Fig. 6 A meeting room for the experiment.

会議開始前には，実験者が被験者に以下のような指示をする．

- 準備稿を直接相手に見せて説明するのではなく，黒板に書き写しながら説明すること．
- 説明のために黒板に記述した図は消さないように黒板上のレイアウトを工夫すること．これによって，以前に説明した部分に関する参照を，その部分の説明以外のときでも自由に行うことをできるようにする．ただし，会議途中で，まだ説明されていないことの先読みはできない．
- 相手に対して疑問や意見などがあれば，その場で述べてよいこと．
- 設計図に関して作業者間に意見の相違があれば，議論を通してそれを解消すること．
- 意見の対立が起こった場合，自分の修正が最小となるように議論をすすめること．
- 2 時間程度で会議は終えてほしいが，とくに制限はない．

なお，会議の参加者は被験者 2 人のみである．

(4) 被験者は会議を行う．

(5) 会議後：会議直後，以下の指示を行う．

- 会議での議論をふまえ，いまここで，自分の担当の図を必要があれば修正し，改定稿を作成すること．
- 自分の担当でない図の方も，白紙から同様に記述し，想像稿の近似を作成すること．

(6) それぞれの被験者は，それぞれに 2 種類の草稿を完成し実験を終える．

3.4 分析対象

実際にはいくつかの補助的な図が描画されたが，実験者が被験者に作成を要求した種類の図のみを分析対象とした．なお，DFD および STD が構造化して記述された場合，構造化された部分を展開することで 1 枚の図にして比較を行う．以下に分析対象として原稿および記録を示す．

準備稿：実験者が準備した OD および RD をもとにして被験者がそれぞれに作成した DFD1 および STD2 (図 5) を用いる．

調整稿：会議の様子をビデオカメラで記録し，それをもとに実験者が記述した原稿を作成する．

改定稿：DFD1′ および STD2′ (図 5) を用いる．

想像稿：想像稿の近似である STD1 と DFD2 (図 5) をそのまま用いる．想像稿の近似は，会議の直後に作成させたため，記憶の劣化による影響は小さいと思われる．

作業過程の記録：調整稿に含まれるそれぞれの要素に関するすべての行為，その時間，そして行為者のデータを用いる．分析対象とする行為は，黒板上での要素に対する記述，削除，指示とする．これらは黒板型計算機 [11] などで自動検出可能なレベルのデータである．

上記のように本実験の記録対象となる行為は，かなり粒度の細かいものである．よって，個々の行為が，2 章で説明した Clark による共通理解を得るための手順における個々のステップもしくはステップの部分に対応すると考えてよい．

また，2 章では ‘認識の不一致’ とみられる部分の中には単なる ‘覚え間違い’ に相当する部分も含まれることも述べた．上記のように本実験での行為は Clark の手順の個々のステップに対応すると考えたため，会議中に，複数回にわたり行為の対象となった部分は，共通理解を成就するための手続きが踏まれたと見なす．よって，会議中にたかだか 1 回しか行為の対象となっていない部分は ‘覚え違い’ であるとして，不一致，一致の分析対象から除外する．

実際には本実験で注目する以外の行為，たとえば ‘頷き’ や ‘目配’ せなどの非言語的行為によって上記の手順が踏まれる場合がある．よって，たかだか 1 回しか行為の対象とならなかった部分を分析対象外とすることは，思い違い’ を除去する基準としては多少厳しいものといえる．たとえば，‘思い違い’ ではなく ‘不一致’ である部分も分析対象外とする恐れがある．しかし，基準が厳しいため，単なる ‘思い違い’ が ‘不一致’ であるとして，分析対象となることはほとんどないため，分析結果の信頼性を維持することができる．

3.5 代案の粒度と抽出法

DFD と STD は構文的には有向グラフとして記述

される．よって始点と終点を共有する部分グラフどうしが互いの代替案の候補となる．その中で，作業過程において時間的に近傍で参照された対，たとえば一方が削除されたうえで他方が記述された場合などは典型的な代替案となる．同様の理由で，始点と終点のどちらかのみを共有している場合でも互いに代替案とする．後述の実験（表 3 と表 4 のデータ）における代案を持つ部分は平均しておよそ 2 個のノードもしくはアークから構成されている．ただし共有する始点および終点は含まない．

4. 結果と考察

表 **1** および図 **7** に実験結果の概要を示す．図表中における数値は生産物の記述言語，DFD および STD のノード数とアーク数に対応する．表 1 中の‘図の種類’の D は DFD，S，S_i は STD を示す．D，S（または S_i）のノード（もしくはアーク）は，図 5 における DFD1′，DFD2 などを用いて表 **2** のように定義される．たとえば，図の種類 D の陰一致のノード（もしくはアーク）とは，図 5 における

$$\overline{\mathrm{DFD1'}} \cap \overline{\mathrm{DFD2}} \cap \text{調整稿}$$

の要素として定義されている．

4.1 カテゴリコードをもとにした不一致の分析

図 7 におけるカテゴリコード 110? の要素は一致する割合が高い．データ数が十分ではないが（カテゴリコード 110? の要素が 5，それ以外が 22），大標本と見なして比率の差を検定すると，$z = -2.46 < -z_{0.05} = -1.645$ となり 5%の有意水準で，110? の要素は不一致となる比率は，それ以外の不一致の比率より小さいといえる．これは，調整中フェーズでの調整の結果，担当者自信が不要と考えた要素は，非担当者も不要であると認識することを意味する．このような理由で，カテゴリコード 110? の要素は一致する割合が高くなると思われる．よって，不一致の発見を行う場合，この領域は重視する必要はないと思われる．

残念ながら，カテゴリコード 110? と 010? のを合わせた部分（陰一致が発生する部分）の不一致となる比率と，それ以外の部分の不一致となる比率は有意な差がなかった．これによって，改定稿に含まれるか否かによって，不一致を判定することは難しいと判断できる．よって，以降，陽一致と陰一致の区別は行わない．

また，カテゴリコード 110? と 111? のを合わせた部分（準備稿に含まれる部分）の不一致となる比率と，それ以外の部分の不一致となる比率も有意な差がなかった．これによって，準備稿に含まれるか否かによって，不一致を判定すること（2.3 節の仮説）も難しいと判断できる．

4.2 代案をもとにした不一致の分析

表 **3**，表 **4** に不一致部分と一致部分それぞれについて，草稿中に発生する部分を，代案の有無をもとにして分類する．代案のない部分に関しては，その内容を調べ以下の 6 つに分類した．

些細：意味的な些細な部分，たとえば図の可読性を向上させるための補助線など．
過渡的：説明などのために過渡的に記述された部分．
慣用句：記述言語の上での典型的な表現法．
当然：事前に与えられた OD，RD に準拠した部分．
新規：新規に決定した部分．
その他：上記以外の部分．

なお，表が煩雑とならないため，部分数が 0 の場合は‘-’と記した．また，割合を計算する際には，上記の‘些細’に分類される部分は除外した．3.5 節で述べたとおり，表 3，表 4 における部分数は，図 7 および表 1 における図中のノードとアークの実数には対応しない．

表 3，表 4，の結果から，不一致部分の代案を持つ比率（74%）は，一致の場合（41%）は比率より大きいように思われる．この場合もデータ数が十分ではないが（不一致の場合の図数が 5，一致の場合が 9），大標

表 **1** 実験結果の概要
Table 1 Results of the experiment.

	ペア#1		ペア#2			ペア#3		ペア#4		
面識	有		有			無		無		
会議時間（分）	50		70			140		81		
図の種類	D	S	D	S_1	S_2	D	S	D	S	計
陽一致	10	16	14	6	7	24	14	20	22	133
陰一致	8	10	7	0	5	19	4	4	8	65
不一致	10	2	14	0	0	7	0	0	5	38
合計	28	28	35	6	12	50	18	24	35	236

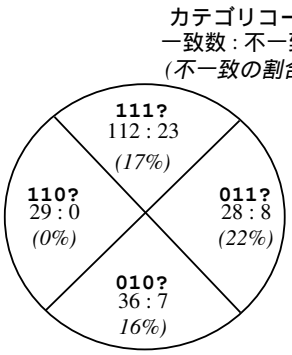


図 **7** カテゴリごとの一致/不一致数
Fig. 7 The distribution of elements.

表 2 表 1 と図 5 の記号の関係
Table 2 Relationships between simbols in Table 1 and simbols in Fig. 5.

表 1 のシンボル	D	S
陽一致	$\mathrm{DFD1}' \cap \mathrm{DFD2} \cap$ 調整稿	$\mathrm{STD2}' \cap \mathrm{STD1} \cap$ 調整稿
陰一致	$\overline{\mathrm{DFD1}'} \cap \overline{\mathrm{DFD2}} \cap$ 調整稿	$\overline{\mathrm{STD2}'} \cap \overline{\mathrm{STD1}} \cap$ 調整稿
不一致	$((\mathrm{DFD1}' \cap \overline{\mathrm{DFD2}}) \cup (\overline{\mathrm{DFD1}'} \cap \mathrm{DFD2})) \cap$ 調整稿	$((\mathrm{STD2}' \cap \overline{\mathrm{STD1}}) \cup (\overline{\mathrm{STD2}'} \cap \mathrm{STD1})) \cap$ 調整稿

表 3 代案の有無をもとにした不一致部分の分類
Table 3 Classifying misunderstood parts with respect to their alternatives.

	ペア#1		ペア#2			ペア#3		ペア#4			
	D	S	D	S_1	S_2	D	S	D	S	計	(%)
代案有	4	2	6	-	-	3	-	-	2	17	(74)
些細	-	-	1	-	-	1	-	-	-	2	-
過渡的	-	-	-	-	-	-	-	-	-	-	(0)
慣用句	-	-	-	-	-	-	-	-	-	-	(0)
当然	-	-	1	-	-	1	-	-	-	2	(9)
新規	-	-	-	-	-	-	-	-	2	2	(9)
その他	-	-	2	-	-	-	-	-	-	2	(9)
合計	4	2	10	-	-	5	-	-	4	25	

表 4 代案の有無をもとにした一致部分の分類
Table 4 Classifying mutual understood parts with respect to their alternatives.

	ペア#1		ペア#2			ペア#3		ペア#4			
	D	S	D	S_1	S_2	D	S	D	S	計	(%)
代案有	6	4	3	-	-	20	2	6	4	45	(41)
些細	1	-	-	-	-	-	-	1	-	2	-
過渡的	-	-	1	-	2	-	-	-	-	3	(3)
慣用句	-	-	-	-	-	-	2	-	-	2	(2)
当然	6	2	6	3	5	10	5	6	6	49	(45)
新規	1	-	-	-	-	-	-	2	5	8	(7)
その他	-	-	1	-	-	2	-	-	-	3	(3)
合計	14	6	11	3	7	32	9	15	15	112	

本と見なして比率の差を検定すると，有意水準 10%では有意な差となりうるが ($z = 1.29 > z_{0.1} = 1.28$)，5%では有意な差はないことになる ($z_{0.05} = 1.645$)．よって，代案が存在することが直接に不一致検出のヒントとはならないため，会議中における不一致が発生している部分の出現分布を調べる．

4.3 調整中フェーズ内での代案どうしの相対的分布

調整中フェーズにおいて，ある部分に関する言及と，その代案に関する部分に対する言及の相対的な分布の特徴を以下の 4 パターンに分類する．

自己残存パターン ：注目する部分に関する言及の方が残存して出現するパターン．

代案残存パターン ：代案の方が残存して出現するパターン．

混在パターン ：注目する部分およびその代案の出現に特別な偏在がないパターン．

不明パターン ：注目する部分，もしくはその代案が領域 ?1?? に含まれないパターン．このパターンでは，代案間の相対的な分布を観測することができない．

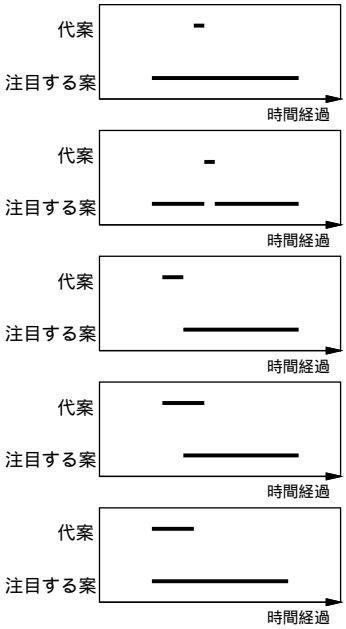


図 8 自己残存パターンの例
Fig. 8 Examples of self-surviving patterns.

図 8 に自己残存パターンの例を示す．第 1 の例は，注目する案が先に発生し，それは継続的に言及されているが，代案はその継続期間のほんの一部分に発生するのみである場合を示している．代案残存パターンの例は，図 8 における ‘代案’ と ‘注目する案’ を逆にしたものとなる．3.4 節で述べたように個々の行為は自動検出可能なレベルのものを用いているため，上記のパターンも自動的に検出することが可能である．表 3，表 4 中の代案を持つ部分について，上記のパターンに基づき表 **5**，表 **6** にその分類結果を示す．

表 6 と表 5 から，‘自己残存または代案残存’ の比率は，不一致に比べ，一致の方が高いように思われる．これもデータ数は十分ではないが (一致が 7 で不一致が 5) だが，大標本と見なして比率の差を検定すると，$z = 1.87 > z_{0.05} = 1.645$ となり 5%の有意水準で，不一致に比べ，一致の方が高いといえる．

‘自己残存または代案残存’ の場合，代案のどちらかが選択され，他方は明示的に棄却されたことを反映す

表 5 不一致部分とその代案との相対出現パターン
Table 5 Classifying the misundertood parts with alternatives with respect to their distribution over the coordination phase.

	ペア#1		ペア#2			ペア#3		ペア#4			
	D	S	D	S_1	S_2	D	S	D	S	計	(%)
自己残存または代案残存	2	-	2	-	-	1	-	-	-	5	(30)
混在	-	2	-	-	-	2	-	-	-	4	(23)
不明	2	-	4	-	-	-	-	-	2	8	(47)
合計	4	2	6	-	-	3	-	-	2	17	

表 6 一致部分とその代案との相対出現パターン
Table 6 Classifying the mutual understood parts with alternatives with respect to their distribution over the coordination phase.

	ペア#1		ペア#2			ペア#3		ペア#4			
図の種類	D	S	D	S_1	S_2	D	S	D	S	計	(%)
自己残存または代案残存	5	4	3	-	-	15	1	4	3	35	(77)
混在	1	-	-	-	-	5	1	-	1	8	(18)
不明	-	-	-	-	-	-	-	2	-	2	(4)
合計	6	4	3	-	-	20	2	6	4	45	

る．よって，認識の一致にいたる部分は，‘自己残存または代案残存’の相対出現パターンをとりやすいことが，上記のような比率差に反映されていると思われる．

それに対して，‘混在’の場合は，どちらかの選択が容易に決まらないことを反映すると思われるため，‘混在’の比率が，不一致の場合の方が大きいと予想できるが，表 6 と表 5 からは有意な差はみられない（$z = 0.24$）．しかも，不一致部分の出現パターンで最も多いのは‘不明’の場合であり，代案どうしの相対的な分布によっては，不一致を直接的に発見することは困難であることになる．

4.4 調整中フェーズで顕在化しない代案

そこで，不明パターンとなった部分の内容を調査した．その結果，不一致となる原因がまったく分からないものもあったが，以下のような特徴が不一致の原因となるとみられるものもあった．

- 代案の部分が非明示的に示されている場合：たとえば図 **9** に示すアーク CA と CB が代案どうしになっているにもかかわらず，アーク CB が調整中フェーズ以前には観測されない場合があった．しかし，アークの終点となっている A と B に関しては，アーク CA の登場と同時に混在パターンとなって出現している．よって，アーク CB が CA の代案となって記述されることを予測できると思われる．
- 代案の部分が変形している場合：たとえば図 **10** におけるアーク ACE とアーク ABE が代案関係にあるにもかかわらず，調整中フェーズにはアーク BE は観測されなかったとする．しかし，アーク BE 上のフローの内容を調べると，アーク AE と一致しており，かつ，調整中フェーズではアーク AE とアーク AE が代案関係であることが分かる．このことより，アーク AE に相当する代案が調整中フェーズ後に発生し，それが不一致を引き起こす可能性を予測できると思われる．
- 担当者の最初の略式説明，言い換えなどに影響されて不一致が発生する場合．
- 部分の詳細度が異なる場合：本実験で題材とした DFD および STD はともに構造的な記述が可能であり，どの程度の詳細な構造までを記述するかは記述者にゆだねられている．よって，あるレベルで文章などの説明で済ませる記述者もいれば，さらに詳細な構造まで当該の言語で記述する記述

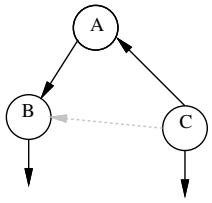


図 9 代案の部分が非明示的に示されている場合の例
Fig. 9 An example of implicit alternative.

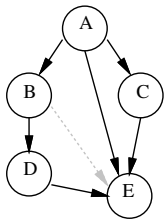


図 10 代案の部分が変形している場合の例
Fig. 10 An example of transformed alternative.

者もいることで不一致が発生する．

5. おわりに

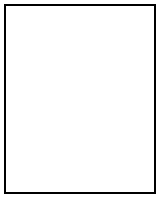
本稿では，作成する生産物の種類が異なるために，認識不一致を直接に発見できないような共同設計作業において，認識不一致が発生が，その設計作業過程にどのように顕在化するかを観測する実験を設計し，その適用を行った．その結果，会議において合意のうえで却下された部分は不一致が少ないことが分かった（4.1 節）．また，代案の存在が若干，不一致の誘引となっている傾向があることが分かった（4.2 節）．ここでの代案関係は図の出現順序，トポロジなどをもとにしているため，意識的に記述を行った場合に書き漏らす恐れのあるものも含まれると思われる．よって，従来の明示的に DR を記述する手法と相補的に利用することが可能であると思われる．会議中における代案どうしの出現分布を調査した結果，一致する部分に関しては採択される代案の方が長命であることが分かった（4.3 節）．実験の施行回数および問題の性質などから，本稿での適用実験の結果が一般的なものとはいえない．しかし，2 章での認識不一致のモデル化と，3 章での実験法によって，分担部分間の関係を直接に調べることが困難な共同作業における認識不一致を検出するための基礎を与えることができたと思われる．

本手法では，担当者と非担当者，注目している案とその代案という形で，つねに 2 対の対立構造を基本とした．しかし，現実には 2 人以上の設計者が参加し，2 つ以上の代案が発生するため，3 対以上の構造を持つ作業のモデル化と認識不一致の定義が必要となる．たとえば，調整段階において折衝を行っている以外の参加者，すなわち傍観者の引き起こす認識不一致も分析する必要がある．

参考文献

1) Aoki, T. and Katayama, T.: Unification and Consistency Verification of Object-Oriented Analysis Methods, *APSEC'98*, pp.296–303 (1998).
2) Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W.: *Object-Oriented Modeling and Design*, Prentice-Hall (1991).
3) Nuseibeh, B., Kramer, J. and Finkelstein, A.: A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification, *IEEE Trans. Softw. Eng.*, Vol.20, No.10, pp.760–773 (1994).
4) Shaw, M. and Garlan, D.: *Software Architechure*, Prentice Hall (1996).
5) Krauss, R.M. and Fussell, S.R.: Mutual Knowledge and Communicative Effectiveness, *Intellectual Teamwork: Social and technological Foundations of Cooperative Work*, chapter 5, pp.111–145, Lawrence Erlbaum Associates (1990).
6) Clark, H.H. and Bennan, S.E.: Grounding in Communication, *Perspectives on Socially shared cognition*, chapter 7, pp.127–149, American Psychlogical Association (1991).
7) Isaacs, E.A. and Clark, H.H.: References in conversation between experts and novices, *Journal of Experimental Phychology: General*, Vol.116, pp.26–37 (1987).
8) Krauss, R.M. and Fussell, S.R.: Perspective-Talking in Communication Representation of Others' Knowledge in Reference, *Social Cognition*, Vol.9, No.1, pp.2–24 (1991).
9) Moran, T.P. and Carroll, J.M.: *Design Rationale Concepts, Techniques, and Use*, Lawrence Erlbaum Associates (1996).
10) Karsenty, L.: An Empirical Evaluation of Design Rationale Documents, *Proc. CHI96*, Vancouver, BC Canada, pp.150–156 (1996).
11) Weiser, M.: Some Computer Science Issues in Ubiquitous Computing, *Comm. ACM*, Vol.36, No.7, pp.75–84 (1993).
12) Clark, H.H. and Schaefer, E.F.: Contributing to Discourse, *Cognitive Science*, Vol.13, No.3, pp.259–294 (1989).
13) Kunz, W. and Rittel, H.: Issues as elements of information systems, Technical Report, S-78-2, Institut fur Gundlagen der Planung I.A. University of Stuttgart (1970).
14) MacLean, A., Young, R.M., Bellotti, V.M.E. and Moran, T.P.: Questions, options and criteria: Elements of design space analysis, *Human-Computer Interaction*, Vol.6, Nos.3 & 4, pp.201–250 (1991).
15) 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 (1984).

(平成 11 年 3 月 30 日受付)
(平成 11 年 10 月 7 日採録)



海谷 治彦（正会員）

1965 年生．1994 年東京工業大学大学院電気電子工学専攻博士課程修了．博士（工学）．1999 年より信州大学工学部情報工学科助教授．ソフトウェア工学，CSCW 等の研究に従事．ACM，IEEE Computer Society 各会員．