

曖昧表現見直しノウハウの形式知化プロセスの提案 —見直しガイドラインの作成と効果検証—

齋藤 忍^{1,a)} 飯村 結香子¹ 山田 節夫¹

概要: ユーザと開発者のコミュニケーションの質の良し悪しは、システム開発の重要成功要因の1つである。システム開発の専門家ではないユーザは、日々の生活で特に意識せずに使われる「程度」を表す曖昧な表現を用いて、開発者にシステムの要求を伝達しがちである。これらの曖昧表現により、システムの仕様を作成する開発者がユーザの要求を捉え違えると、ユーザの要求に即さないシステムが実装され、作り直しを招き、納期遅延やコスト増加に至る。開発者には、ヒアリング等を通じて得られたユーザの要求について曖昧表現がないかを確認し、曖昧な点について再ヒアリング等を通して解消すること（曖昧表現の見直し）が求められる。そこで本稿では、曖昧表現の見直しを支援するノウハウとして、曖昧である理由、曖昧表現の見直しの方針、見直し後の例文、そして例文を書く上で考えるポイントの4つを定義し、ノウハウを形式知化するプロセスを提案する。その上で、提案プロセスを適用し形式知化されたノウハウ（曖昧表現の見直しガイドライン）の効果を被験者実験により検証し、提案プロセスの有用性や課題を示す。

Knowledge Formalization Processes for Addressing Ambiguity Words in System Requirements

1. はじめに

企業・組織のシステム開発では、システムを利用するユーザからの要求に基づき、開発者がシステムの仕様を策定する。そのため、ユーザと開発者のコミュニケーションの質の良し悪しは、システム開発の重要な成功要因の1つである [2]。一方、システム開発の専門家ではないユーザは、日々の生活で特に意識せずに使われる「程度」を表す曖昧な表現を用いて、開発者にシステムの要求を伝達しがちである [15]。これらの曖昧表現により、システムの仕様を作成する開発者がユーザの要求を捉え違えると、ユーザの要求に即さないシステムが実装される。結果として、システムの作り直しを招き、納期遅延やコスト増加に至る。

システムの仕様を作成する開発者は、ユーザへのヒアリング等を通して得られた要求について、要求内容を確認し、曖昧な点を発見し、曖昧な点については再ヒアリングなどを通して解消する、ことが求められる。これら曖昧表現の見直し (=発見+解消) の作業は、これまでは経験者に依存

するケースが多かった。例えば文献 [12] では、ウォーターフォール型の開発スタイルを採用しているプロジェクトに対して、システムの仕様を作成後に品質ゲート（第3者の経験者によるインスペクション）を設けることで、曖昧表現を含むシステムの仕様が後工程（設計・実装工程）にすり抜けることを防止した取り組みを報告している。しかしながら、アジャイル型の開発スタイルを採用するプロジェクトが増加し、より迅速なデリバリーが求められる現状では、各開発者が曖昧表現の見直しを自ら実施することが求められる。これまでに筆者らは、経験の豊富な技術者のノウハウが、曖昧表現の見直しに役立つことを報告している [13]。一方、ノウハウをどのように形式知化するかに関する議論は不十分であった。

そこで本稿では、曖昧表現の見直しを支援するノウハウとして、曖昧の理由、曖昧表現の見直しの方針、見直し後の例文、そして見直しするうえで考えるポイントの4つを定義し、これらノウハウを形式知化するためのプロセスを提案する。次に、企業の開発プロジェクトで作成された要求定義書に対して、提案プロセスを適用し、曖昧表現の見直しノウハウの形式知化（見直しガイドラインの作成）を行う。その上で、形式知化されたノウハウの効果を被験者

¹ 日本電信電話 (株)
NTT Corporation, Chiyoda, Tokyo 100-8116, Japan
^{a)} shinobu.saitou.cm@hco.ntt.co.jp

実験により評価し、提案プロセスの有用性を示す。被験者実験は、システム開発の経験の浅い技術者（初心者）、および経験の豊富な技術者（経験者）の2種類の被験者群に対して行う。被験者実験では、以下に記す2つの研究仮説を設定し、その検証を試みる。

- 仮説1：提案プロセスは、初心者が曖昧表現を発見・解消する上で有効なノウハウが形式知化できる。
- 仮説2：提案プロセスは、経験者が曖昧表現を発見・解消する上で有効なノウハウが形式知化できる。

本稿の構成は以下のとおりである。2章では、関連研究を述べる。システムの仕様の曖昧さに関する研究を概観する。3章では、本研究で対象とする「程度」を表す曖昧な表現を示し、曖昧表現見直しのノウハウを形式知化するための5つのプロセスを提示する。4章では、提案プロセスの適用により形式知化したノウハウ（曖昧表現の見直しガイドライン）の内容を示す。5章では、システム開発の初心者・経験者に実施した実施した、見直しガイドラインの有効性評価の実験の内容を示す。6章で考察を行い、最後の7章で結論と今後の課題を述べる。

2. 関連研究

2.1 システム要求における曖昧表現

ソフトウェア要求仕様に対する推奨プラクティスを記したISO 830 [4]では、良いソフトウェア要求仕様書の特性として「正確」や「完成」、「一貫」とならび「無曖昧」が挙げられている。ISO830をリプレースしたISO/IEC/IEEE 29148:2018[5]では、個別の要求の満たすべき特性として「無曖昧性」を挙げており、言語表現上の基準として漠然とした、あるいは一般的な語は避けるべきとして曖昧な語例を示している。

ソフトウェア開発を対象とした知識体系としてはSWE-BOK[10]が知られている。近年、要求定義を含む上流工程に特化した技法やプロセスを体系化した知識体系も提唱されている[6],[7]。特に要求工学に関する知識体系のREBOK[7]は、SWEBOKを補完することを狙いとしており、11の要求の特性（単一性、完全性、一貫性、法令順守、独立性、追跡可能性、最新性、実現可能性、無曖昧性、必要性、検証可能性）を提唱している。そのなかの「無曖昧性」は、文字通り「曖昧さがないことであり、2つ以上の解釈がないこと」と定義されている。

システム要求の品質と、プロジェクトの最終的な成否には相関関係があることは以前より報告されている[1]。Wiegiersら[14]は、プロジェクトが失敗する一般的な原因として、ステークホルダの関与不足や要求のクリープと並び、曖昧な要求を挙げている。Masseyら[9]は、規制(regulations)に関する要求の曖昧さに着目し、それらに起因するプロジェクトの問題を指摘している。

2.2 曖昧表現の発見と解消

文献[3],[15]では、システム要求の文章を検索し、予め登録された、曖昧さを招き易い用語（キーワード）を検知する技法を提案している。これらの技法ではキーワードを予め抽出・整備しておくことが求められる。Wiegiersら[14]も、曖昧さの一因となる語句の一覧を挙げている。システム要求の文章からキーワードや語句を機械的に検知することで、曖昧表現の発見には一定の効果が期待できる。しかしながら、曖昧さの見直しの難しさは、上述のアプローチ（単純な文字の検知）では曖昧表現の解消までは至らないことである。他の開発者にシステム要求の文章を回覧し、コメントをもらうだけでも、曖昧表現の見直しには不十分である[14]。曖昧表現を含む文章は、回覧先（開発者）により異なる理解（解釈の違い）が発生した状態となる。各々の理解に基づくコメントが指摘されるに留まり、曖昧表現の解消（一意に意味が限定された状態）には至らない。

Pohl[11]は、自然言語で記述された要求の曖昧さを5つに分類している（表1参照）。さらに、曖昧さ（ある文章に対する2つ以上の解釈）を避ける方法として、用語集、要求構文パターン、制約言語を挙げている。用語集を作成し、文書中で利用する語の意味を限定すれば、lexicalな曖昧さを避けることができる。要求構文パターンや制約言語のように文章の構文、さらには使用する用語を限定することでsyntactic, semanticな曖昧さを避けることができる。ただし、統制が強く自然言語としての表現力の低い制約言語は要求獲得とよばれるシステム仕様の策定の初期には向かないと述べている。文を短くする、能動態で書く、「そして」、「また」などの複数の文をつなげる接続詞を避けるといった記述スタイルの規定もsyntactic, semanticな曖昧さの排除に繋がる[14]。“referential”, “vagueness of terms”については、文献[11]では曖昧さの分類として挙げるに留まり、文献[5],[14]では使用を避けることのみが示されている。ただし、“referential”については、指示語・代名詞を使わない代わりに名詞を再記述する、あるいは、指示語・代名詞に名詞を補足することで解釈を限定する方法が知られている。“vagueness of terms”は、日常的に多く使用されており、書き手も読み手も意味はとれてしまうため曖昧

表1 曖昧さの分類

Table 1 Types of ambiguity.

曖昧さの分類	説明
Lexical	ある単語に異なる別の意味が存在する。
Syntactic	文章に複数の構文解釈（妥当な統語関係）が存在する。
Semantic	文章の意味の解釈が複数存在する。
Referential	文章中の参照先の解釈が複数存在する。
Vagueness of terms	文章中に外延的な定義が一般的に不明か定義が困難な「表現」が含まれる。

な表現であることが見逃されやすい。また、個々の曖昧表現をどのように見直すべきかは開発者（見直す人）の知識や経験に依存しがちである。従って、曖昧表現の解消も、他の分類の対処法（文章の形式的な変換）と比較して難しい。そこで、本稿における見直しの対象とする曖昧表現は、曖昧さの分類”vagueness of terms”とする。

3. ノウハウの形式知化プロセス

3.1 ノウハウの活用シーン

曖昧表現の見直しには、システム要求の文書から曖昧表現を発見することも重要であるが、発見された曖昧表現には「どのような情報が足りないか、あるいは不明であるか」を開発者自身が考えることも重要なポイントとなる。その上で、それまでのユーザへのヒアリング内容、もしくは追加のヒアリングを行うことで情報を補い、曖昧表現の書換え（解消）を行う。このような見直しに向けた一連のステップを実施するには経験者のノウハウが有用である。

本稿が想定する、形式知化されたノウハウの活用シーンを図 1 に示す。以下、活用シーンにおけるユーザと開発者とのステップの概要を記す。

- ステップ (1) 要求の提示：ユーザは、開発対象システムの要求を開発者に提示する。
- ステップ (2) 曖昧表現の発見：開発者は、ユーザからのシステムの要求に含まれる曖昧表現を発見する。
- ステップ (3) 修正案の作成：開発者は、発見した曖昧表現に対して、修正案（書換え後の表現）を作成する。
- ステップ (4) 確認：開発者は、作成した修正案（書換え後の表現）をユーザに確認する。

以上の 4 つのステップを、開発者とユーザの間で必要に応じて繰り返し実行し、システムの要求から曖昧表現を解消する。形式知化されたノウハウはステップ (2) と (3) の開発者の作業を支援する。

3.2 形式知化の方針

本稿における見直しの対象とする曖昧表現”vagueness of

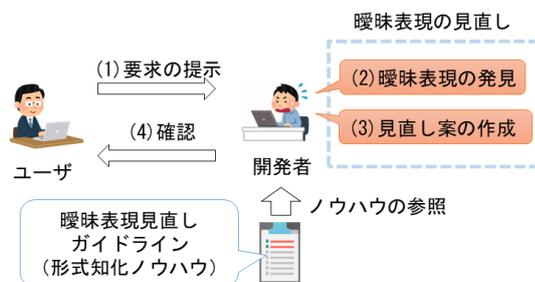


図 1 形式知化ノウハウの活用シーン

Fig. 1 Use case image on formalized knowledge for addressing ambiguity words in system requirements.

terms” の例としては”Fred is tall”が知られており、”tall”が曖昧表現に相当する。この文章の書き手は、Fred が属するクラス（年齢や参加している競技、職業など）、およびそのクラスにおける身長の高い低いを判断する基準を想定した上で、Fred の身長が「高い」と表現している。文章中にはその基準が示されないため、読み手は自身の経験などから身長が「高い」と判断する基準を想定し「高い」と言う表現を解釈する。書き手、読み手それぞれは自身が想定する基準で「高い」を解釈できてしまう。書き手と読み手の間で基準が異なり得ることで見逃されやすい。”vagueness of terms”は一般的にも広く利用されている。”vagueness of terms”による曖昧さへの対処はその表現を使わないことのみ解説されることが多いが、例えばユーザがある”vagueness of terms”を用いて要求を表現した際には、その表現が複数の解釈をとりうる理由を理解した上で、複数の解釈例から、ある一つの解釈になるようにユーザから不明な情報を引き出さなくてはならない。”vagueness of terms”の数は多く、それぞれの表現ごとにどのような解釈が可能かは異なる。そこで、本稿では、実際の要求文書から個々の”vagueness of terms”を抽出し、それぞれについて、複数の解釈が生じる理由、曖昧表現の見直しの方針、見直し後の例文、見直しする上で考えるポイント、の 4 つを経験者のノウハウとして定義し、ガイドラインとして形式知化することとした。次節では、これらのノウハウを形式知化プロセスの内容を示す。

3.3 形式知化プロセスの内容

曖昧表現の見直しノウハウを形式知化するための 5 つのプロセスを規定した（図 2 参照）。形式知化プロセスは、曖昧表現抽出と見直し案作成の 2 つのフェーズに分けられる。2～5 番目のプロセスの実施結果が、形式知化されたノウハウとして書換えガイドラインの各構成要素に反映される。見直し案作成フェーズは、曖昧表現の抽出フェーズの実行者全員で曖昧表現の個数分だけ繰り返し実施する。図 3 にプロセスとガイドラインの構成要素の対応関係を示す。以降では各プロセスの実施内容を記す。

(1) 抽出元の選定

実際に開発プロジェクトで使われていた曖昧表現を抽出するため、過去のプロジェクトで作成された要求定義書を選定する。選定の観点としては、次のプロセスで抽出する曖昧表現の種類の偏りを少なくするため、異なる業界・業種に向けて開発した複数の要求定義書を選定することが望ましい。

(2) 曖昧表現の抽出

前プロセスで選定された要求定義書から複数人の開発経験者により、前節の形式知化の方針に基づき、曖昧表現とそれを含むシステム要求の文章の抽出を行う。経験者には「読み手（設計者）により複数の解釈が発

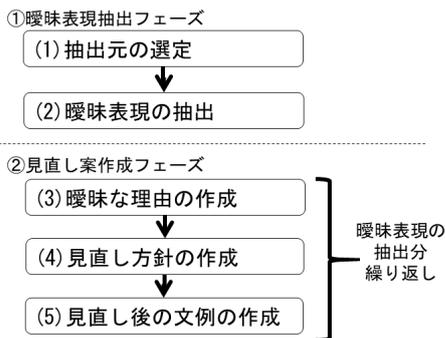


図 2 曖昧表現見直しノウハウの形式化プロセス

Fig. 2 Knowledge formalization processes for addressing ambiguity words in system requirements.

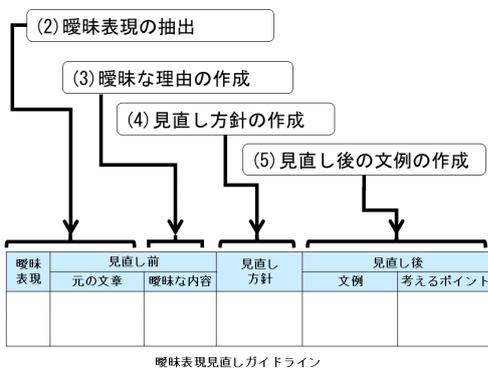


図 3 プロセスとガイドライン（形式化されたノウハウ）の構成要素の対応関係

Fig. 3 Mapping relation between knowledge formalization processes and elements of guideline.

生し得る程度表現」を曖昧表現の判定基準として伝える。一方、特定のドメイン用語などは曖昧表現の対象とはしない。その上で、曖昧表現、およびそれらの表現を含む文章（要求定義書から抜き出した文章）を、図 3 に示すガイドラインの対応する構成要素（読み、曖昧表現、元の文章）に記載する。なお、同じ曖昧表現を含む文章が複数抽出された場合は全ての文章を記載する。

(3) 曖昧な理由の作成

抽出された曖昧表現を含む文章のそれぞれに対して、なぜ曖昧であるのかの理由を定義していく。具体的には、当該文章では、読み手（設計者）が要求内容を具体化（仕様化）していく上で、開発対象のシステムの機能や振舞いが一意に定まらないことを、曖昧である理由として規定していく。作成された理由は、図 3 に示すガイドラインの対応する構成要素（曖昧な内容）に記載する。

(4) 見直し方針の作成

抽出された曖昧表現に対して、どのような観点に着目すれば、曖昧さの解消ができるのか（見直しの方針）

を検討・定義する。定義された方針は、図 3 に示すガイドラインの対応する構成要素（見直し方針）に記載する。

(5) 見直し後の文例の作成

元の文章（曖昧表現を含む要求文章）に含まれていた曖昧表現の見直し後の例文を作成する。併せて、例文を作成する上で、どのように思考して曖昧表現を解消したのか（考えるポイント）を、例文に即して具体的に記載する。これら 2 つの内容を、図 3 に示すガイドラインの対応する構成要素（文例、考えるポイント）に記載する。

4. 形式化プロセスの実践

4.1 形式化プロセスの適用

曖昧表現抽出フェーズの (1) 曖昧表現の選定では、企業（システムインテグレータ）の業務システム開発で実際に作成された要求定義書 9 個を選定した。選定にあたり、記述内容の偏りを無くするため、異なる業界・業種の顧客に向けて開発したシステムの要求定義書を対象とした。9 つの開発プロジェクトは、当該企業内で全て異なる組織で実施しており、全てのプロジェクトはウォーターフォール開発のアプローチを採用していた。

選定した要求定義書は、全て新規開発のプロジェクト時に作成されたものとした。この理由は、記述項目の網羅性確保のためである。システムの機能拡充（追加・修正等）や更改のプロジェクトで作成される要求定義書は、過去に作成された（当該システムの新規開発時の）要求定義書に対する差分のみ記載される場合が多い。調査対象として、十分な要求文書を含む要求定義書を選定するため、機能拡充や更改の要求定義書は候補から外した。

以上の 9 つの要求定義書に対して、執筆者ではない開発経験者（第 3 者）3 名により曖昧表現の抽出フェーズの (2) 曖昧表現の抽出、合計 395 個の曖昧表現を含む文章を抽出をした。最終的に文章中の重複を除いた合計 91 個の曖昧表現を抽出した。

表 2 に抽出された曖昧表現のサンプルを 4 つ示す。表の左から 1 列目は、曖昧表現が記されている。2 列目は、その曖昧表現が実際の要求定義書でどのような文の中に含まれていたのか（元の文）を示している。3 列目は、なぜその表現が読み手にとって解釈が一意に決められない（曖昧な内容）になるのかの説明となる。抽出された曖昧表現の品詞別の割合は、名詞が 6 割強で最も多く、次が形容詞の 3 割弱となり、残りが副詞であった。

次の見直し案作成フェーズでは、91 個の曖昧表現に対する、見直し前、見直し方針、見直し案（文例と考えるポイント）の 3 つのノウハウを、3 人の開発経験者の議論により定義した。見直し前については、曖昧表現抽出フェーズで抽出した「元の文」、「曖昧な内容」について元の文章が

表 2 曖昧表現の例

Table 2 Examples of vague terms in System Requirements.

曖昧表現	元の文章	曖昧な内容
明らか	画面 Y に表示の画像が明らかである場合、画像表示確認のダイアログを表示しない。	どのような状態が「明らか」かが分からない。
難しい	業務 Y の中で難しい作業はシステム化対象外とする。	「難しい」が該当する状況が分からない。
特に	特に大きいファイルの場合、送信は不可とする。	「特に」の誇張する程度が分からない。
近隣	検索に当たったユーザの近隣ユーザに関するデータも表示する。	「近隣」が指す範囲が分からない。

案件の知識がないと理解しにくい場合には書き換えを行った。最終的に、91 個の曖昧表現に対しては、21 個の重複を含む 70 個の見直し案を作成した。

4.2 形式知化されたノウハウの内容

作成された曖昧表現の見直しガイドラインは、抽出された曖昧表現に対して、見直し前、見直し方針、見直し後、の 3 つの内容で構成されるリストの集合体 (= 91 行のリスト) である。図 3 に、表 2 で示した 4 つの曖昧表現のそれぞれに対する見直しのノウハウの内容を示す。

● 見直し前

曖昧表現がどのような曖昧さを持っているのか (何が悪いのか?) に関する内容が、ガイドラインの 2 列目「元の文章」と 3 列目「曖昧な内容」に記されている。記載内容は、前節のプロセス 2 とプロセス 3 の実施結果に基づく。例えばリスト「特に」では、3 つの場合 (3 行分) が記載されている。これは曖昧表現「特に」を含む文章が、前節のプロセス 2 において 3 種類が抽出されており、それぞれに対してノウハウを作成したことを意味している。

● 見直し方針

曖昧表現を解消する上で何を明確化すべきであるか (見直し時に心がけることは?) に関する内容が、ガイドラインの 4 列目「見直し方針」に記されている。記載内容は、前節のプロセス 4 の実施結果に基づく。一方、リスト「特に」の 3 番目の場合では、見直しは不要と記載されている。これは、たとえ曖昧表現として抽出されたとしても、見直しが不要の場合があるという経験者のノウハウが形式知化された一例である。

● 見直し後

曖昧表現を解消する上で何を指すべきか (具体的なお手本は?) に関する内容が、ガイドラインの 5 列目「文例」と 6 列目「考えるポイント」に記されている。記載内容は、前節のプロセス 5 の実施結果に基づいている。あくまで見直しの一例ではあるが、曖昧表現の

書換えを実施した際の経験者の意図を言語化している。

5. 被験者実験

提案プロセスにより形式知化されたノウハウ (曖昧表現の見直しガイドライン) の効果を検証するため、被験者による比較実験を 2 つ実施した。1 つ目はシステム開発の経験が浅い被験者 (初心者) を対象とした実験、2 つ目は経験が豊富な被験者 (経験者) を対象とした実験である。どちらの比較実験でも、被験者を、曖昧表現見直しリストを提供しない集団 (グループ A) と、提供する集団 (グループ B) の 2 つのグループに分ける。グループ A の被験者には曖昧表現の一覧を配布し、グループ B の被験者に対しては、加えて見直し前、見直し方針、見直し後を含むガイドラインを配布する。両グループに曖昧表現の発見・解消を問う問題に解答してもらい、形式知化されたノウハウの提供有無に伴う両グループの正答率の結果を比較する。

5.1 実験の被験者

● 初心者

ソフトウェア工学の授業を履修していた 80 名の大学生を無作為に 2 つに分けた。全ての大学生は同じ大学・同じ学部にも所属していた。一方を「初心者グループ A」とし、他方を「初心者グループ B」とした。

● 経験者

企業 (システムインテグレータ) に所属しソフトウェア開発経験年数が 10 年以上である 20 名の技術者を無作為に 2 つに分けた。全ての技術者は同じ企業に所属している。一方を「経験者グループ A」とし、他方を「経験者グループ B」とした。

5.2 実験の問題

全ての被験者は、要求文書のなかから曖昧表現を発見する問題と、曖昧表現を解消する問題の 2 つに解答する。曖昧表現の発見問題は 15 問、解消問題は 11 問で構成される。発見問題の内訳は、8 問がガイドラインに記載の表現、2 問がガイドラインに未記載の表現である。残りの 5 問は、曖昧表現が問題文中に存在しない (即ち、曖昧表現は無いが正答となる)。解消問題の内訳は、8 問がガイドラインに記載の表現、3 問がガイドラインに未記載の表現である。

5.2.1 曖昧表現の発見問題

図 4 に、曖昧表現の発見問題において、実際に被験者に出された問題と正答の内容を示す。各問には、実際のシステム開発の要求定義書で使われた文章が記載されている。文章は曖昧表現を含んでおり、図の例では「適正」が曖昧表現となる。被験者は問題の文を読み、文中に含まれる曖昧表現を発見する。

5.2.2 曖昧表現の解消問題

図 5 に、曖昧表現の解消問題において、実際に出された

表 3 曖昧表現の見直しガイドラインの内容 (抜粋)

Table 3 Example of Guideline.

曖昧表現	見直し前		見直し方針	見直し後	
	元の文章	曖昧な内容		文例	考えるポイント
明らか	画面 Y に表示する画像が明らかである場合、画像表示確認のダイアログを表示しない。	どのような状態が「明らか」かが分からない。	「明らか」となる対象と、「明らか」にする手段の双方を記載する。	画面 Y に表示する画像がアクタ A により指定されたものである場合、画像表示確認のダイアログを表示しない。	明らかにする手段を具体的に示すため、下線部の表現に書換え。
難しい	業務 Y の中で難しい作業はシステム化対象外とする。	「難しい」が該当する状況が分からない。	何にとつてどのよう「難しい」のかを記載する。	業務 Y の中で人手が介在せざるを得ない作業はシステム化対象外とする。	システムによる完全自動化が難しい状況を明確にするため、下線部の表現に書換え。
特に	[他と比べて程度を誇張する場合] 特に大きいファイルの場合、送信は不可とする。	「特に」の誇張する程度が分からない。	「特に」が指す量的情報を記載する。	100 MB を超えるファイルの場合、送信は不可とする。	ファイルの大きさを明確にするため、定量的な値を含む下線部の表現に書換え。
	[他との区別を示す場合] 業務 A については、特に処理 A が必要となる。	「特に」がどういう意味で特別なのか分からない。	他と区別する情報を記載する。	業務 A については、他の業務にはない処理 A が必要となる。	処理 A がどういった意味で「特に」なのかを明確にするため、下線部の表現に書換え。
	[「特になし」と記載の場合]	-	見直しは不要	-	-
近隣	検索に当たったユーザの近隣ユーザに関するデータも表示する。	「近隣」が指す範囲が分からない。	どこまでを「近隣」と示すのかを指す量的情報を記載する。	検索に当たったユーザの DB 上の前後 5 レコード (当たったユーザは含まない) があるユーザに関するデータも表示する。	「近隣」の範囲を明確にするため、定量的な値を含む下線部の表現に書換え。

問題文

客席のタッチパネルまたは接客係のハンディターミナルで受けた注文の飲料や料理を、一品一葉の受注伝票として適正に発行する。なお、一品一葉とは1つの伝票に1品のみ書くことをいう。

「適正」は曖昧な名詞

■ 曖昧な内容

何にとつてどうすることが「適正」かが分からない

図 4 曖昧表現の発見問題と正答例

Fig. 4 Example of Question and Correct Answer on Correcting Ambiguity Words in Experiment.

問題と正答例の内容を示す。各問の文章中には、予め曖昧表現の出現箇所に下線が引かれており、被験者が分かるように指定されている。図の例では「事前」が曖昧表現となる。被験者は問題文を読み、文中の曖昧表現の解消を行う。被験者は実際に要求定義書が作成された当時の執筆者ではないため、当時のユーザのシステム要求の内容は把握していない。そのため被験者には、指定された曖昧表現について、不足する情報は仮想して補った上で解消するように求めた。

問題文

発注された書籍に貼付するラベルと挿入する貸出カードを事前印刷する。

正答例

発注された書籍に貼付するラベルと挿入する貸出カードを納入予定日のX日前(まで)に印刷する。

採点基準:

日(・時)を特定できていれば正答。
 「発注時に」「発注日の翌営業日」は可だが、「納入される前に」「貸し出す前に」は不可。

図 5 曖昧表現の解消問題と正答例

Fig. 5 Example of Question and Correct Answer on Correcting Ambiguity Words in Experiment.

発見問題の採点とは異なり、解消問題では予め用意した正答との単純な比較で採点(正誤判定)することができない。そのため、解消問題の採点では、企業において要求定義書の執筆・レビューの実施した経験のある3人の開発者を、採点者として用意した。採点者の3人は、4章で述べたガイドラインの作成に関わった3人とは異なる。各問題の採点においては、3人の採点者が解答(被験者が見直し

た文章)を読み、事前に準備した正答例・採点基準と照らし合わせて、曖昧表現が排除されているかを各々が判断する。なお、曖昧さが解消されているか否かのみを問うものとし、要求文として適切かは問わないこととした。例えば、図5の「事前」は、日時が特定できるように見直されていれば正解とし、「発注日の翌日」のように営業日の考慮がないなどは正誤判定に影響しない。その上で、3人の採点結果を照らし合わせ、個々の解答(被験者が見直した文章)に対して、3人の採点者の全員が曖昧さが解消されていると判断した場合のみ、当該解答は正解であると判定した。従って、1人でも採点者が、曖昧さが残る文であると判断すれば、当該解答は誤答であると判定した。

5.3 実験の手順

2つの比較実験は同様の手順で実施した。

手順1: 両グループに対して、要求定義書における曖昧表現について解説する(約30分)。この際、ガイドラインに記載の曖昧表現(91個)の一覧のみ、説明資料に記している。

手順2: グループAを実験会場から一旦退席させ、グループBにのみガイドラインを配布し、その見方を説明する(約15分)。

手順3: グループAを再び実験会場に戻した上で、曖昧表現の発見問題と解消問題の用紙を両グループに配布する。各被験者に問題への解答をしてもらう。(約60分)。

手順4: 採点者が解答結果を採点した上で、両グループの正答率の比較を行う。

5.4 実験の結果

5.4.1 曖昧表現の発見の実験結果

初心者の2つのグループ、及び経験者の2つのグループの正答数のばらつきを図6に示す。

● 初心者

2グループの正答数の平均は、「初心者グループA」(ガ

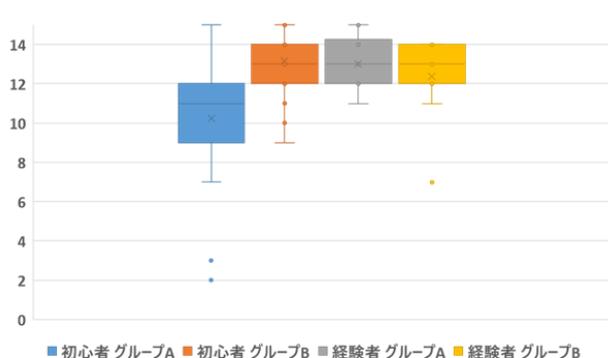


図6 曖昧表現の発見における正答数の比較(箱ひげ図)

Fig. 6 Results of Identifying Ambiguity Words for Four Participants Groups (Box chart).

イドライン無)が10.2点であり、「初心者グループB」(ガイドライン有)は13.2点となった(15点満点)。ガイドライン有のグループの平均値は、ガイドライン無のグループの平均値より高くなった。次に2グループの平均点の差の検定を実施した。等分散の検定(F検定)の結果を踏まえ、t検定(等分散を仮定した2標本の検定)を実施した結果、両グループの平均点の差に5%水準で優位差が認められた。

● 経験者

2グループの正答数の平均点は、「経験者グループA」(ガイドライン無)が13.0点であり、「経験者グループB」(ガイドライン有)は12.4点となった(15点満点)。初心者の実験結果とは異なり、ガイドライン無のグループの平均点が、ガイドライン有のグループの平均点より高い結果となった。次に2グループの平均点の差の検定を実施した。等分散の検定(F検定)の結果を踏まえ、t検定(分散が等しくないと仮定した2標本の検定)を実施した結果、両グループの平均点の差には5%水準で優位差は認められなかった。

5.4.2 曖昧表現の解消の実験結果

初心者の2つのグループ、及び経験者の2つのグループの正答数のばらつきを図7に示す。

● 初心者

2グループの正答数の平均点は、「初心者グループA」(リスト無)が5.7点であり、「初心者グループB」(ガイドライン有)は7.8点となった(11点満点)。ガイドライン有のグループの平均点は、ガイドライン無のグループの平均点より高くなった。次に2グループの平均点の差の検定を実施した。等分散の検定(F検定)の結果を踏まえ、t検定(等分散を仮定した2標本の検定)を実施した結果、両グループの平均点の差に5%水準で優位差が認められた。

● 経験者

2グループの正答数の平均は、「経験者グループA」(ガイドライン無)が8.4点であり、「経験者グループB」(ガイドライン有)は9.5点となった(11点満点)。初心者の

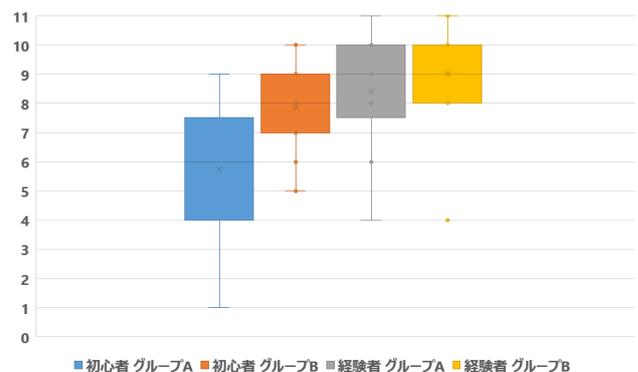


図7 曖昧表現の解消における正答数の比較(箱ひげ図)

Fig. 7 Results of Correcting Ambiguity Words for Four Participants Groups (Box chart).

実験結果と同様に、ガイドライン有のグループの平均点は、ガイドライン無のグループの平均点より高くなった。次に2グループの平均値の差の検定を実施した。等分散の検定(F検定)の結果を踏まえ、t検定(分散が等しくないと仮定した2標本の検定)を実施した結果、両グループの平均点の差には5%水準で優位差は認められなかった。

5.5 仮説の検証

前節の比較実験の結果に基づき、本稿の1章で提示した2つの仮説の検証を行う。

5.5.1 仮説1の検証

前節の「初心者」に対する実験の結果より、形式知化されたノウハウ(曖昧表現の見直しガイドライン)は、システム開発の経験が浅い初心者にとって、要求定義書内の曖昧表現を正しく発見し、且つ正しく解消するために有効であることが統計的に実証された。以上より、1章で提示した仮説1の内容が正しいことが確認できた。

5.5.2 仮説2の検証

前節の「経験者」に対する実験の結果より、形式知化されたノウハウ(曖昧表現の見直しガイドライン)は、システム開発の経験が豊富な技術者に対して、要求定義書内の曖昧表現の発見と解消に有効であることを統計的に実証するに至らなかった。そのため、1章で提示した仮説2の内容が正しいことは確認できなかった。

6. 考察

6.1 形式知化プロセスの有用性

本稿における曖昧表現の見直しとは、要求定義書に記載された文書の中で、後続工程の設計者やプログラマが読んだ際に解釈が一意に定まらない表現(曖昧表現)を発見し、その上でユーザの要求に基づき、曖昧表現を解消する一連の作業を意味する。曖昧さを見出すこと、解消することはどちらも難しい。本稿では複数の開発経験者のノウハウは、曖昧表現の発見・解消の双方に有効であるとの考えのもと、ノウハウを形式知化するプロセスを提案し、実践を行った。実験結果より、システム開発の初心者が形式知化されたノウハウを参照することで、曖昧表現を見つけること、および解消することに効果があることが確認できた(正答率の差に統計的に優位な差が現れた)。即ち、提案プロセスは、初心者に対して曖昧表現の発見・解消に有用となるノウハウを形式化できることが実証できた。

一方、企業で10年以上のソフトウェア開発の経験を有する技術者(経験者)に対しては、曖昧表現の発見と解消のどちらについても形式知化されたノウハウの有効性(正答率の向上)は確認できなかった(統計的に優位な差としては現れなかった)。しかしながら、実験に参加した経験者からは、「曖昧表現の解消を感覚的に行っていたが、具体的な例で何が曖昧になる原因かを示されており改めて理解

できた」などの意見もあった。初心者で見られた有意な効果(正答率の向上)は確認できなかったが、提案プロセスは、経験を積んだ技術者も有用と感じられるノウハウを形式知化できたといえる。

6.2 初心者と経験者の違い

曖昧表現の発見と解消の実験結果の内容について、初心者と経験者の結果を対比させながら議論する。初めに曖昧表現の発見では、ガイドラインを提供された初心者グループBの平均点は、経験者の2グループの平均点と同等の結果となった。曖昧表現の発見においては、初心者でも複数の開発者のノウハウを参照することで、経験者の作業水準(曖昧表現の発見の正答率)に到達できている。次に曖昧表現の解消では、ガイドラインを提供された初心者グループBの平均点は、ガイドラインを提供されなかった経験者グループAの平均点に及ばなかった。初心者には、たとえ複数の開発者のノウハウを参照することができたとしても、経験者の作業水準(曖昧表現の解消の正答率)に達することは困難であった。前述の曖昧表現の発見とは異なり、初心者が曖昧表現の解消できるようになるには、学習(ノウハウの参照)だけではなく、訓練(OJT等)も組み合わせる必要があるといえる。

6.3 形式知化プロセスの課題

本稿で述べた提案プロセスの実践により、9つの要求定義書から91個の曖昧表現を抽出した。プロセスの適用対象の要求定義書を増やすことで、抽出される曖昧表現は増やすことが可能である。ただし、形式知化されたノウハウが含む曖昧表現の数を増やす(曖昧表現の網羅性を高める)ことだけが、ノウハウの有用性を高める唯一の手段ではないと考える。抽出した個々の曖昧表現やその見直しノウハウについて、グルーピングや構造化などにより、抽象度の高いノウハウを抽出するステップの追加が有用であると考ええる。また、本稿では、利用者の観点(例:曖昧表現の一覧性や検索容易性)からの評価や検証も実施していない。利用者からのフィードバックに基づきノウハウを改善するプロセスを追加することも必要であると考ええる。

7. おわりに

本稿では、ユーザのシステム要求における曖昧表現を見直し(発見・解消)するため、開発者(経験者)のノウハウを形式知化するプロセスを提案した。企業で作成された要求定義書を対象に提案プロセスを適用し、ノウハウの形式知化(曖昧表現の見直しガイドラインの作成)を実施した。被験者による比較実験により、提案プロセスは、システム開発経験の浅い技術者(初心者)が曖昧表現の発見と解消をする上で有用なノウハウを形式知化できることを検証できた。今後の課題は、プロセスの適用やノウハウの参

照の作業を支援（自動化）するツールの整備が挙げられる。

参考文献

- [1] Barney B.R.: 5.1.4 The Benefits of Integrated, Quantitative Risk Management, INCOSE International Symposium, vol. 11, no. 1, pp. 120-125 (2001).
- [2] Davis M. A.,: Just Enough Requirements Management: Where Software Development Meets Marketing, Dorset House (2005).
- [3] 久野綾子, 平尾英司, 五藤智久: 仕様書の曖昧性を検出するツールの試作と評価, 電子情報通信学会, 電子情報通信学会総合大会講演論文集, vol. 2012, no. 1, pp. 27(2012).
- [4] IEEE Std. 830-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE (1998).
- [5] ISO/IEC/IEEE 29148: 2018(E), International Standard - Systems and software engineering - Life cycle processes - Requirements engineering (2018).
- [6] IIBA: BABOK: A Guide to the Business Analysis Body of Knowledge, Lightning Source Inc (2015).
- [7] 一般社団法人情報サービス産業協会, REBOK 企画 WG : 要求工学知識体系, 近代科学社 (2011).
- [8] 河野哲也, 猪塚修, 藤森麻紀子, 本間周二, 茂中義典: キーワードベースドレビューードキュメントのあいまいさや不備に着目したレビュー手法 - (online), 入手先 (<http://jasst.jp/archives/jasst10e/pdf/C2-3.pdf>) (2020.07.08).
- [9] Massey A., Rutledge L., Antón A., and Swire P.: Identifying and classifying ambiguity for regulatory requirements, 2014 IEEE 22nd International Requirements Engineering Conference (RE), pp. 83-92 (2014).
- [10] 松本吉弘 (翻訳): ソフトウェアエンジニアリング基礎知識体系 - SWEBOK V3.0 -, オーム社 (2014).
- [11] Pohl, K.: Requirements Engineering: Fundamentals, Principles, and Techniques, Springer Publishing Company, Incorporated (2014).
- [12] 斎藤忍, 竹内睦貴, 山田節夫, 青山幹雄: RISDM: ソフトウェア要求仕様書のインスペクションデザイン方法論の提案と適用評価, ソフトウェアエンジニアリングシンポジウム 2014 論文集, pp. 105-114 (2014).
- [13] 斎藤忍, 飯村結香子, 山田節夫: 要求文書における曖昧表現の見直し, JISA Quarterly, No. 132, 2019 Winter(2019).
- [14] Wiegers K. E. and Beatty J.: Software Requirements, 3rd edition, Microsoft Press (2014).
- [15] 山本芳郎, 高山啓: 曖昧キーワードチェックツールの開発と実践的適用 (online) 入手先 (<http://www.juse.jp/sqip/library/shousai/?id=137>) (2020.07.08).