

修正候補検出を目的とした 時相論理に基づく要求仕様書の分析

永田 拓也^{1,a)} 山本 椋太¹ 吉田 則裕¹ 高田 広章¹

概要：

組込みシステムの開発の現場では、開発対象のシステムの仕様を開発前に決定し要求仕様書にまとめる。要求仕様書は自然言語で記述されることから、文書内に誤り・曖昧表現を含む場合が考えられる。関連研究に、要求仕様書に対して形態素解析システムおよび日本語格・照応解析システムによる分析を加え状態遷移モデルの抽出支援を行う研究がある。この研究では、要求仕様書中の文を字句・構文解析によって節へ分割され3種類の分類がなされた。本研究では、この状態遷移モデルを抽出支援するツールの中間出力を用いた手法を提案する。提案手法では時相論理における曖昧表現を修正候補とし、要求仕様書中に含まれる時相論理に関する記述に着目して分析を行うことで、修正候補を検出を行う。提案手法において、時相論理を含む自然言語が記述できる形式言語を定義した。修正候補を検出を行うために、要求仕様書中に現れる時相論理に関係すると考えられる単語を抽出し辞書作成を行なった。作成辞書は、日本語ソーラス辞書を活用し類似の意味を持つ単語を含む。本手法の適用実験を自動車向け組込みシステムに対して実施した。修正候補の正解集合と提案手法による抽出数の比較を行った結果、要求仕様書中の特定の節に対して手法の効果を確認した。

Analyzing a Requirements Specification Document Based on Temporal Logic for Identifying Correction Candidates

TAKUYA NAGATA^{1,a)} RYOTA YAMAMOTO¹ NORIHIRO YOSHIDA¹ HIROAKI TAKADA¹

1. はじめに

組込みシステムの開発の現場では、開発対象のシステムの仕様を開発前に決定し要求仕様書にまとめる。要求仕様書は自然言語で記述されることから、文書内に誤り・曖昧表現を含む場合が考えられる [7][10]。このような誤り・曖昧表現に基づいた開発を行うことで、手戻りが発生することがある。開発の手戻りを低減するには、要求仕様書に含まれる誤り・曖昧表現をシステムの設計を行う前に、修正する必要がある。

本研究では、要求仕様書に含まれる誤り・曖昧表現を自動検出し分析を行う手法の提案を行なう。誤り・曖昧表現の中でも、システムの実現の方法に関わる時相論理に関する

記述に着目し、分析を加える。この分析により、時相論理に関する記述において曖昧な表現を検出し、修正候補として検出することを目的とする。提案手法を用いるために、時相論理に関する記述を辞書として登録し検出された結果は、構造的なデータ形式へ格納を行なった。辞書登録した単語は、日本語ソーラス辞書 [1] を参照し、対象とする組込みシステムの要求仕様書に現れる状態遷移を表す単語とした。

2. 関連研究

2.1 要求仕様書から状態遷移モデルの抽出

中村らは、組込みシステムを対象として要求仕様書から状態遷移モデルの抽出を行う研究を行なった [6]。自然言語で記述された要求仕様書に対して、提案手法を適用し解析木の作成およびユーザとインタラクティブにやり取りをす

¹ 名古屋大学
Nagoya University, Nagoya, Aichi 464-8603, Japan
^{a)} t-nagata@ertl.jp

表 1 節の分類条件 (文献 [6] 中の表 4 より引用)

条件節	節の末尾が副詞的名詞 + (格助詞—接続助詞—副助詞—読点) である 例: 以下の条件が成立している場合、
処理節	直前の節が条件節である 例: (条件節) + スイッチを「オン」にする。
定義節	文中に条件節が存在しない 例: 車は「停止中」「走行中」の いずれかの値を取る。

ることで、状態遷移表の抽出支援を行った [5]。中村らの手法では、日本語形態素解析システムとして JUMAN*¹および、日本語文・格・照応解析システムとして KNP*²を利用し、解析木の作成を行う。要求仕様書の 1 文を入力とし、意味のまとまりによって節へ分類する。節は、それぞれ条件節、処理節、定義節へと分類された。各節への分類は次の表 1 の基準で行われた。

表 1 について説明する。条件節は、KNP の出力結果を用いて判定を行う。処理節および定義節は、自身の条件節判定の結果および直前の節の結果を用いて判定を行う。

この研究で提案された手法を実現するために、状態遷移モデル抽出支援を行うツール (以下、状態遷移抽出ツール) が実装された [5]。状態遷移抽出ツールでは、入力された 1 文ずつの自然言語文に対して、JUMAN および KNP を適用し、解析木の作成をしている。この解析木は、箇条書きの文を除き、1 文に対して 1 つの解析木が作成され、1 つのテキストファイルへ格納されている。

1 つのテキストファイル中に、要求仕様書中の 1 文が節に分割され、表 1 に従って分類されたノードが存在する。それぞれのノードでは、JUMAN および KNP を適用した結果の一部の情報を持つ。要求仕様書中の全ての文に対してこの解析を行った結果、出力されるテキストファイル群が、状態遷移抽出ツールが持つ中間出力である。

中村らは、状態遷移抽出ツールに要求仕様書を適用する前に前処理のために次の a1 から a5 の処理が行われた [5]。これらの処理は、JUMAN が持つ制約に基づく処理である。a1 から a3 の処理は計算機によって自動的に行われ、a4 および a5 の処理は、中村らによって手動で実施された。

- a1 文末が『ため。』の文を解析対象から除去する。
- a2 入力文の半角文字を全て全角文字に変換する。
- a3 アルファベットの単位を日本語に変換する。
- a4 数式の文頭にメタ文字を付与する。
- a5 要求仕様を 1 文単位に分割する。

状態遷移抽出ツールの中間出力は、次の b1 から b9 に記述されている情報を持っている。形態品詞情報および品詞細分類情報は、JUMAN が出力する結果に含まれる情報で

ある。形態品詞情報は、名詞や助詞といった形態素を分類する情報である。品詞細分類情報は、形態品詞情報をさらに細分化した情報であり名詞に対して固有名詞、助詞に対して接続助詞のような分類が与えられる。

- b1 解析木中でのノード ID
- b2 形態素解析を行った結果である形態品詞情報
- b3 形態品詞情報を細分化した、品詞細分類情報
- b4 解析した節の日本語文
- b5 解析木中での親ノードの ID
- b6 解析木中での子ノードの ID
- b7 解析木中でのノードの位置情報
- b8 ノードの分類情報
- b9 実装上の情報

本研究では、要求仕様書に対する解析を行う際にこの中間出力を利用する。ゆえに、本研究で扱う自然言語文は、前処理および JUMAN, KNP の適用がなされた文章であると言える。

また、我々の研究グループでは、状態遷移抽出ツールの節の分類の精度の調査を行った [11]。文献 [11] によれば、ある要求仕様書に対して節分類の調査を行った結果、precision は 0.9 を超えており、分類を誤った記述は箇条書きの中に複数の節が存在するものだけであった。本稿では、4 章にて述べるが、文献 [11] において対象とした要求仕様書を用いる。本稿では今後、節の分類は正しく行われている前提で議論する。

2.2 時相論理を活用した要求仕様記述言語

Li らは、時相論理を活用した要求仕様記述言語として Temporal Action Language (以下、TeAL) と呼ばれる言語を開発した [10]。TeAL では、要求仕様を表す自然言語文から、時相論理に関わる表現を抽出し、形式言語で記述することができる。TeAL は次のように、シグネチャ、シグネチャ接続表現、時相論理候補の 3 つの構成要素から成り立つ。

- シグネチャは、システムを構成する 1 つ 1 つの要素の属性の表現。
- シグネチャ接続表現は、シグネチャ間に成り立つ関係の表現。
- 時相論理候補は、シグネチャ接続表現に対して時間的な制約の表現。

このとき、シグネチャおよびシグネチャ接続表現は、時相論理候補に関する情報を持たない。

加えて Li らは、自然言語で記述された要求仕様文から TeAL 表現を自動的に抽出する手法の研究を行なった [4]。この手法は、自然言語文から TeAL 表現への変換を計算機が支援するものである。手法の手順は、次の 4 ステップに分かれている。

Step1 計算機が自然言語文を入力として受け取り、要求

*1 <http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN>

*2 <http://nlp.ist.i.kyoto-u.ac.jp/index.php?KNP>

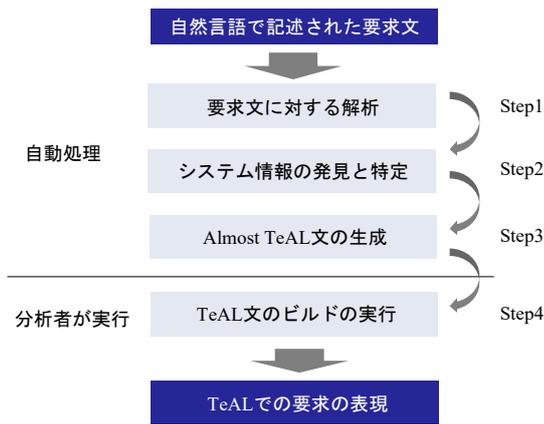


図 1 TeAL 生成の手順 (文献 [4] の Fig.1 を参考に作成)

文に対して形態素解析を行う。

Step2 計算機がシステムの詳細を表す情報を抽出し、特定する。

Step3 計算機が、解析結果、および特定したシステム情報を元に自動で TeAL 文を生成する。これを Almost TeAL 文という。

Step4 人間が手動で、計算機によって生成された Almost TeAL 文を修正する。

これらの処理を図に表すと、図 1 のようなフローとなる。この研究により、計算機が自動的な処理を行い、自然言語で記述された要求仕様を記述した文から、形式的な言語への変換を支援することが確認された。本研究では、TeAL の構文規則を利用し、日本語で記述された文を形式的に表現可能な言語を検討する。

2.3 要求仕様書中の誤り・曖昧表現の分類

位野木らは、一貫性に注目して要求仕様の一貫性検証知識の形式知化や要求仕様の一貫性検証の支援ツールの開発およびその評価を行った [2], [3]。彼女らは、実際に企業から検証知識をインタビューによって抽出して共通部分、可変部分に仕分け、検証ルールと辞書として形式知化した。また、これらの検証ルールと辞書を組み込んだ要求仕様の一貫性検証支援ツールを開発した。彼女らが提案したツールは、要求仕様書の一貫性として用語の不一致に着目し、省略表現、修飾表現および表記揺れに着目している。

竹内らはテキスト分析技術を用いた開発関連文書の品質分析を行った [8]。組込みシステムのソフトウェア開発で作成された要求仕様書、アーキテクチャ文書、設計文書を対象として、自然言語処理を用いて曖昧表現の抽出を試みている。この文献 [8] においては、ガ格 (主語) が省略されている問題が多く見つかっており、文献 [8] では、文全体のうちおよそ 50% 程度存在した。加えて、受動態表現が、トピックとなる語句を見つける際の妨げになる可能性についても言及している。

山本らは、組込みシステム開発で利用されている要求仕

様書に存在する誤り・曖昧表現の実証的調査を行った [12]。この調査では、要求仕様書中の誤り・曖昧表現を 9 つの問題に分類した。分類した問題に対して自動検出する手法を検討した。自動検出への検討の結果、辞書を使用するか字句・構文レベルの解析によって検出可能な問題は解析が比較的容易であるとし、読み手が文意を解釈可能な場合が多い問題は、文意を解釈不可能な場合のルール定義が必要であるとまとめた。

要求仕様書中の誤り・曖昧表現の中に、方法の欠如 [12] という分類がある。方法の欠如では、制限事項の不足としてデットラインを含む非機能要件が提示されていないことが例として挙げられている。山本らの研究ではこの問題に対して詳細な分類を行っていない。そこで本研究では、方法の欠如の一部である時相論理に関連する曖昧な記述を抽出を試みる。

3. 提案手法

本章では、時相論理に関連する記述のうち、修正候補となる記述を自動検出する手法について述べる。本手法では前処理として、要求仕様書中に現れる単語のうち、時相論理に関係する記述と考えられる候補を手動で抽出した。抽出した単語と類似の意味を持つ語の網羅性を高めるため、日本語シソーラス辞書 [1] を用いて類義語の検索を行った。検索で見つけた単語のうち、元となる要求仕様書中に現れる単語を選択し、時相論理に関係する語として辞書作成した。

次に、2.1 節で述べた状態遷移抽出ツール [5] の中間出力に対して、作成辞書の適用を行い修正候補の検出を試みた。本手法の詳細について、3.1 節から具体的に述べる。

3.1 線形時相論理

古典命題論理の拡張として時間の変化で真偽値が動的に変化する時相論理がある [9]。時相論理のモデルのうち、線形時相論理では、表 2 のような命題式および演算子が与えられる。

これを図式化すると、図 2 のようになる。この図において、 ϕ, ψ はそれぞれ真偽値をとる命題論理式である。 \square は、各時刻を表し論理式が記述されている場合は、その論理式が真であることを示す。また、各行において、記述されている命題論理式が全て真の時、時相論理式 $X\phi, F\phi, G\phi, \phi U\psi, \phi R\psi$ が真となる。

3.2 要求仕様書を解析する形式言語

本研究の目的を達成するために、要求仕様書を分析する形式言語を定義した。この形式言語は次の図 3 中の式 3.1 から式 3.3 を含む言語である。

この言語を構成する要素について説明する。 $SENTENCE$ は、要求仕様書中に現れる句点、および

表 2 線形時相論理における文字の定義

表記	分類	意味
ϕ	命題式	真偽値が与えられる命題式
ψ	命題式	真偽値が与えられる命題式
$X\phi$	単項演算子	ϕ が次状態で ϕ が真ならば、 $X\phi$ が真となる時相論理式
$G\phi$	単項演算子	未来の全ての状態で ϕ が真ならば、 $G\phi$ が真となる時相論理式
$F\phi$	単項演算子	未来のある時点で ϕ が真ならば、 $F\phi$ が真となる時相論理式
$\phi U \psi$	二項演算子	ψ が真となる直前の状態まで ϕ が真になるとき、 $\phi U \psi$ が真となる時相論理式
$\phi R \psi$	二項演算子	ϕ が真となる状態まで ψ が真になるとき、 $\phi R \psi$ が真となる時相論理式

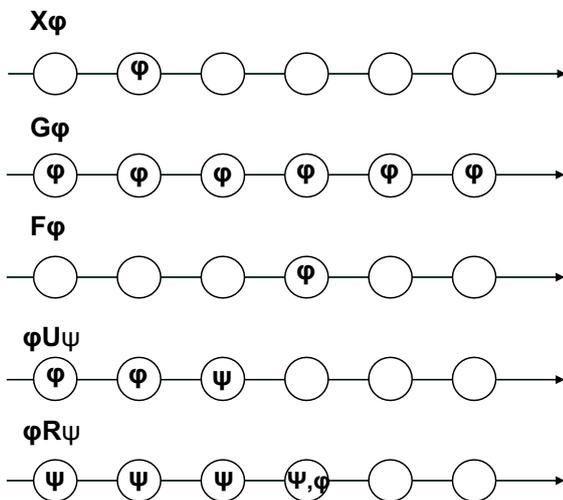


図 2 線形時相論理式が真となる時を表す図 (文献 [9] 中の第 10 章 2.1 節の図を参考に作成)

$$\begin{aligned}
 \text{SENTENCE} &= \{\text{CLAUSE}, \text{AD}\}^+ && \text{式3.1} \\
 \text{CLAUSE} &= \{\text{CLAUSE}_E, \text{CLAUSE}_P, \text{CLAUSE}_D\} && \text{式3.2} \\
 \text{CLAUSE} ::= (S, AC, TC) &&& \text{式3.3}
 \end{aligned}$$

図 3 要求仕様書を解析する形式言語

箇条書きで区切られる文字列を表す (以降、これを文と呼ぶ) AD は、節末尾にあって、次の節との関係性を表す語句、または節先頭にあって、前の節との関係性を表す語句を表す。 S は、節中の主語を表す。 AC は、主語の動作を表す。 TC は、節中に現れる時相論理に関する記述を表す。

3.1 節で述べた線形時相論理と、要求仕様書を解析する形式言語の対応づけを説明する。式 3.2 の $CLAUSE_D$ は、定義節であることからシステム上で考えられるどの時刻においても真であるべきと考える。ゆえに、線形時相論理上における G (Globally) に対応させて表すことにする... 式 3.2 の $CLAUSE_E$ および、 $CLAUSE_P$ は、それぞれ条件

節と処理節であり、システムのある時点において真となる。ここで処理節は、条件節成立後の次に実行される処理として対応づけする。これは線形時相論理上における $X(\text{neXt})$ と定義づける。

条件節は、節単体ではその前の時点が定義されていないと考え、線形時相論理における F (Future) に対応づけて考える。しかし、条件節は状態遷移抽出ツールによって条件節と条件節が接続される場合と、条件節の後に処理節が接続される場合が存在する。条件節と条件節が接続される時の後方の条件節は、処理節と同様に $X(\text{neXt})$ と定義づけを行う。

2 項の命題式に対する線形時相論理 U (Until), R (Release) については、各節に現れる状態に対して、同じ状態に関する記述を同定し、2 つの状態の間にどのような状態が存在するか解析を行う必要がある。この解析手法については、本手法の対象外とする。

3.3 時相論理における修正候補

本研究の目的とする要求仕様書中に現れる修正候補の例を要求仕様書の例 1 を用いて説明する。

要求仕様書の例 1

- 文 1 電源スイッチ信号が「オン」の状態が 100 ミリ秒継続した状態から、「オフ」に変化した場合、電源スイッチ押下判定を「オン」とする。
- 文 2 電源スイッチ信号が「オフ」または、電源スイッチ信号が「オン」の状態が 100 ミリ秒継続しなかった状態から、「オフ」に変化した場合、電源スイッチ押下判定を「オフ」とする。
- 文 3 電源スイッチ状態が「電源オフ」の時、電源スイッチ押下判定が「オン」の場合、電源スイッチ状態を「電源オン」にする。
- 文 4 電源スイッチ状態が「電源オン」の時、電源スイッチ押下判定が「オン」の場合、電源スイッチ状態を「電源オフ」にする。
- 文 5 電源スイッチ状態は「電源オン」、「電源オフ」のいずれかの値をとる。
- 文 6 電源スイッチ押下判定は「オン」、「オフ」のいずれかの値をとる。

要求仕様書の例 1 における文 1 において、「電源スイッチ信号が『オン』の状態が 100 ミリ秒継続した状態から『オフ』に変化した場合」が条件節となり、「電源スイッチ押下判定を『オン』とする」が処理節となる。同様に、文 4 において「電源スイッチが『電源オン』の時、」および、「電源スイッチ押下判定が『オン』の場合」が条件節となり、「電源スイッチ状態を『電源オフ』にする」が処理節に該当する。この場合において、文 1 中で処理節の「電源スイッチ押下

判定を『オン』にする」状態は、文 4 中の条件節に対応する。このようにして、要求仕様文の例から次の状態 **S1** から状態 **S7** を抽出することができる。また、状態 **S1** から状態 **S7** は真偽値を判定することのできる論理式といえる。

S1 電源スイッチ信号が「オン」の状態が 100 ミリ秒継続した状態から、「オフ」に変化した状態

S2 電源スイッチ押下判定を「オン」とする状態

S3 電源スイッチ押下判定を「オフ」とする状態

S4 電源スイッチ信号が「オフ」である状態

S5 電源スイッチ信号が「オン」の状態が 100 ミリ秒継続しなかった状態から「オフ」に変化した状態

S6 電源スイッチ状態が「電源オン」である状態

S7 電源スイッチ状態が「電源オフ」である状態

この時、状態 **S1** から状態 **S7** および 3.1 節にある線形時相論理演算子を利用することで、要求仕様書の例 1 は、次のように記述 1 から記述 6 に対応した記述ができる。

記述 1 $S1 \rightarrow X(S2)$ が真となる

記述 2 $S4 \vee S5 \rightarrow X(S3)$ が真となる

記述 3 $(S7 \rightarrow S2) \rightarrow X(S6)$ が真となる

記述 4 $(S6 \rightarrow S2) \rightarrow X(S7)$ が真となる

記述 5 $G(S6 = \neg S7)$ が成り立つ

記述 6 $G(S2 = \neg S3)$ が成り立つ

状態 **S1** は、「電源スイッチ信号が『オフ』に変化した」という記述を含み、**S1** が真の時、**S4** も真である。この時、「 $S1 \rightarrow X(S2)$ 」および、「 $S4 \vee S5 \rightarrow X(S3)$ 」が同時に真となる。このとき、「 $G(S2 = \neg S3)$ 」に反する。

これは、要求仕様文の例 1 中の文 1 において、次の 3 つの状態を合わせ持つことが原因となり発生する。

- 電源スイッチ信号が「オン」の状態
- 現状態を 100 ミリ秒維持した状態
- 電源スイッチ信号が「オフ」の状態

3.4 修正候補の検出手順

本手法では節 3.2 の形式言語に含まれる式 3.3 において、抽出される TC (時相論理に関係する記述) について解析を行う。このため、要求仕様書を分析する形式言語のうち式 3.3 において、抽出される S や AC の節中の主語やその動作を表す表現の抽出は本手法の対象外とする。本手法における修正候補とは、節を 1 つの状態とした時に他の節から検出される状態を複数組み合わせることができる節とする。節を 1 つの状態とした時に、節が示す状態の粒度が統一されれば、節間でどの状態を示しているか解析を行うことができる。これにより異なる節間および節中において同じ状態を検出し、初期状態からどのような経路が存在するか解析を行うことができる。よって、節毎の真偽値を線形時相論理で記述できると考えられる。

修正候補の検出は、以下の手順によって実施する。この手順を図式化したものが、図 4 である。図 4 において、手

作業で実施した箇所が手順 **A** および手順 **B** に該当する。自動で実施した箇所が手順 **C**、手順 **D** および手順 **E** である。

手順 **A** 元の要求仕様書から時相論理に関わると考えられる単語を手作業で抽出を行う。元の要求仕様書から時相論理に関わると考えられる単語を手作業で抽出する。時相論理に関わると考えられる単語は、3.3 節で記述の修正候補と考えられる条件を満たす。

手順 **B** 手順 **A** で抽出した単語を日本語シソーラス辞書 [1] で検索し、関連する単語の辞書登録を行う。手順 **A** にて抽出した単語に対して、日本語シソーラス辞書 [1] にて検索を行う。この検索により要求仕様書から手作業で偶然抽出した単語と類似の意味を持つ単語を辞書登録し、要求仕様書中から検出すべき単語の網羅性を上げるために行う。このうち辞書には、検索された単語のうち、元の要求仕様書中で発見される単語のみを登録するものとする。

手順 **C** 要求仕様書の文書を句点もしくは箇条書きで分割したファイルを状態遷移抽出ツールへ適用する。状態遷移抽出ツールは、要求仕様書中の句点および箇条書きで分割されたテキストファイルを入力とする。このテキストファイルに対して、処理を行い 2.1 節で述べた状態遷移抽出ツールの中間出力が得られる。この中間出力は、入力された句点および箇条書きで分割された 1 文に対して、解析木を 1 つ作成したものである。

手順 **D** 中間出力が持つ情報を取得し、条件節 ($CLAUSE_E$)、処理節 ($CLAUSE_P$)、および、定義節 ($CLAUSE_D$) 別にテキストファイルへ格納する。手順 **C** で作成された解析木から条件節、処理節、および定義節が存在するノードを抽出し、条件節、処理節、定義節に分けて節の集合を作成する。節の集合別にファイルを作成し格納する。

手順 **E** 手順 **D** で分類された節に対して、辞書に基づいた解析を行う。解析後は、結果を出力して終了する。節 1 つ 1 つに対して辞書登録された単語の有無を確認し、辞書登録された単語を含む節に対してラベル付けを行う。このように解析した結果を構造的データに格納して出力する。

3.5 実装

状態遷移抽出ツールの中間状態を入力し、修正候補を出力して終了する一連の手順のうち、3.3 節の手順 **A** および手順 **B** は、要求仕様書中からの単語の抽出、および日本語シソーラス辞書での検索、結果の辞書登録であり手作業にて実施した。3.3 節の手順 **C**、手順 **D** および手順 **E** については、Python プログラムとして実装した。まず時相論理に関する記述として辞書登録する処理について説明し、次にプログラムで実装した手順について説明する。

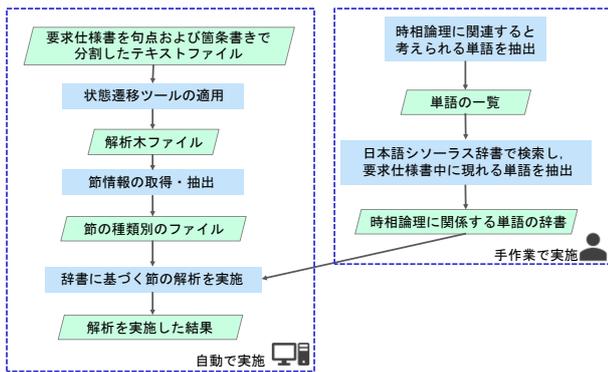


図 4 提案手法の手順

3.5.1 手作業で行う処理

手順 A において抽出する単語は、節中に状態の変化を含む表現を抽出すると考えられる。要求仕様書の文章を目視で確認しこのような表現を探した。この作業により、要求仕様書から時相論理に関する記述として変化、遷移の 2 単語を抽出した。

次に、手順 B では、手順 A で抽出した単語を日本語シソーラス辞書 [1] にて検索し、類語の検出を行った。これにより、要求仕様書中に現れた単語を抽出すると、切り替えの 1 単語であった。

以上から前処理は、変化、遷移、切り替えを時相論理に関わる単語として辞書登録を行った。辞書は、辞書を登録するテキストファイルにカンマ区切りとして保存した。

3.5.2 実装したプログラムの詳細

実装したプログラムは、次の手順で構成されている。

手順 a まず、状態遷移抽出ツールの中間出力をテキストファイルで読み込む。状態遷移抽出ツールの中間出力は 1 文に対する解析木が 1 つのファイルに格納されている。次の出力例は、解析木の 1 つのノードに対応する。テキストファイルでは、このノードが 1 行ずつ記述されている。

出力例

```
subtree_midasi0_10['subtree_midasi0_10',((名詞+助詞)+(名詞+助詞)), '(サ変名詞+接続助詞)+(副詞的名詞+格助詞)', 'オンの時に', 'sentence_midasia_0', ['subtree_midasi0_5', 'subtree_midasi0_3'], '節', '条件節', 2, -1, 0, 0, '-1']
```

読み込んだ文字列データの先頭に、そのデータを格納していたファイル名を付け加え、解析木がどの文と対応しているか追跡できるように情報を付加する。

手順 b 手順 a で出力されたテキストファイルを 1 行ずつ読み込む。読み込んだ入力データから節ノードのみを抽出する。このとき、節ノードが持つ、条件節・処理節・定義節の情報を利用して、条件節のみ、処理節の

み、定義節のみをそれぞれ集約したテキストファイルへ、節ノードを格納する。

手順 c 分類された節ごとに、辞書ファイルを参照し、時相論理に関係する単語の探索を行う。条件節ファイル、処理節ファイル、定義節ファイルから 1 行ずつ読み込み、節が、辞書登録された単語を含んでいないか登録辞書に基づいて検索を行う。辞書登録された単語のうち、少なくとも 1 つ含む場合は、時相論理に関係する単語を含む節とする。

手順 d 時相論理に関係する単語が存在する場合は抽出する。手順 c で、時相論理に関係する表現を含む節と認識された節に、手順 b で節を格納していたテキストファイル名の情報付加し出力用変数へ格納する。

手順 e 手順 d の処理が終了すると、「節があった元のテキストファイル名」、「節の分類」、「節の本文」、「修正候補であるか」の情報が明らかになる。これらの情報を整形し実装したプログラムの出力結果として CSV ファイルへ書き込みを行い、処理を終了する。

csv ファイルの出力例

```
textfile000,条件節,電源スイッチの信号が「オン」である状態が 100 ミリ秒継続した状態から「オフ」に変化した時、修正候補
```

4. 適用実験

本適用実験では、4.1 節で述べる実験対象に対して提案手法を適用し、修正候補の自動検出を実施する。自動検出の結果から適合率 (*precision*) および、再現率 (*recall*) を計測する。また、*precision* と *recall* の調和平均を *F* 値とする。これらの計測結果から提案手法に対する考察を加える。

4.1 実験対象

実際に企業で使用されている自動車向け組込みシステムの要求仕様書を本実験の適用対象とする。この要求仕様書に対して、2.1 節で記述された前処理が実施された。前処理を実施し状態遷移抽出ツールの適用を行った結果、状態遷移抽出ツールの中間出力を格納した 243 個のテキストファイルを得られた。このテキストファイルを本適用実験の適用対象とする。

4.2 実験手順

適用実験は次の記述する手順に沿って実施する。

手順 α 提案手法中の手順 D で節の種類でまとめられたファイルごとに、手作業で修正候補を持つ節の集合を作成する。1 節ずつチェックし状態を分割すべきと考えられる節を正解集合とした。

手順 β 提案手法中の手順 E を実施し、実装したプログラ

表 3 節ごとの集計数

節分類	正解集合数	自動検出数	正解集合に含まれる 自動検出の数
条件節	18	15	13
処理節	8	3	0
定義節	1	1	1
合計	27	19	14

表 4 precision, recall, F 値

節分類	precision	recall	F 値
条件節	0.87	0.72	0.79
処理節	0.00	0.00	0.00
定義節	1.00	1.00	1.00

ムを適用し、自動検出によって得られる修正候補を得る。これにより次の出力例に記載のような出力を得た。

出力例

textfile000, 条件節, コントロールスイッチの信号が「オン」である状態が 100 ミリ秒以上継続した場合から「オフ」に変化した場合, 修正候補

手順 γ 手順 α で作成した正解集合, および手順 β で検出した自動検出数を集計し, precision, recall および F 値を次の式に従って計算した。

$$precision = \frac{\text{正解集合に含まれる自動検出の数}}{\text{自動検出による検出数}}$$

$$recall = \frac{\text{正解集合に含まれる自動検出の数}}{\text{正解集合の数}}$$

$$F = \frac{2 * precision * recall}{precision + recall}$$

上述した手順のうち, 手順 β はツールによって自動的に処理を行った。手順 α および手順 γ は, 手作業で実施した。

4.3 実験結果

4.2 節によって記述された手順により, 次の表 3 および, 表 4 の結果を得られた。表 4 の各値の計算結果は, 小数点第 3 位を四捨五入して計算した結果である。条件節, 処理節, 定義節の各ファイルの母数は, 条件節は 113 個, 処理節は 86 個, 定義節は 32 個であった。

4.4 考察

まず, precision について考察する。条件節では, 他の節と比較して高くなった。これは, 3.4 節の手順 A で時相論理に関わると考えられる記述として登録した単語と関連があると考えられる。この時に「変化」や「遷移」という条件節に現れる単語を登録したため, 結果の値が高くなったと考えられる。また, 日本語シソーラス辞書 [1] で検索した単語から追加した「切り替え」は正解集合に含まれておらず, precision の値を下げる結果となった。処理節では,

値が 0.00 となった。例えば「システム電源を『オン』から『オフ』にする。」というように「変化」や「遷移」という表現がないためと考えられる。自動検出数によって検出された 3 つの節は, 日本語シソーラス辞書 [1] で検索した単語から追加した「切り替え」によって検出されたものである。

次に, recall について考察する。条件節では, 「継続した場合」のように動作の完了を意味する表現が正解集合に含まれた。辞書作成の際にこのような表現を登録できなかった。このことから, recall の値が precision より小さくなったと考えられる。処理節では, 「から」のように時間上の変化を表す表現であったものの, 助詞であったため辞書作成の際にこのような表現を登録できなかった。加えて, 処理節において「変化」や「遷移」のような単語を利用して時間上の変化を表すと「変化させる」や「遷移させる」という表現になる。これは受動態を用いることから直接的で分かりやすい表現ではないと考える。このため, recall の値が 0.00 になったと考えられる。

定義節では, 定義節への分類条件から時間上での変化が存在することは考えにくい。このため, 正解集合数が少なくなったと考えられる。定義節中で検出された唯一の節は, 時間上の変化を伴う制約表現であった。

本適用実験の妥当性への脅威について考察する。自動車向け組込みシステムの要求書 1 つのみを対象としており, 異なる要求仕様書では異なる結果が得られる可能性がある。また本手法は, 時相論理に関連がある単語を要求仕様書中から抽出し実装を行っているため, 初期にどのような単語を抽出・登録するのかによって実験結果が大きく左右されると考えられる。さらに正解集合を第 1 著者自らが作成しており, 複数回チェックを行うなど作成への注意を払ったが, この点で異なった結果となる可能性が考えられる。

4.5 提案手法の限界

提案手法では, 節を時相論理における 1 つの時点とした時に他の節を複数組み合わせた表現となる節を修正候補とし検出を行った。この分類の他に時相論理における曖昧な表現として, 前後を表す表現やシステム全体の中での時間を表す表現が考えられる。本研究ではこのような事例に対する手法の検討を行っていない。

5. まとめ

本研究では時相論理に基づく要求仕様書から修正候補を検出する手法について提案した。また提案手法の実装を行い, 要求仕様書から修正候補を検出する手順の自動化を行った。適用実験の結果, 修正候補の自動検出ができることを確認した。

本研究の目的は, 既存研究で分類された要求仕様書に含まれる誤り・曖昧表現に着目し, 時相論理に基づく分析を加えることで, 修正候補の検出を行うことであった。適用

実験を実施した結果、要求仕様書から修正候補の検出を行えることを確認した。よって、今後はその他の修正候補を検出し修正することができる。その後、条件節、処理節、定義節の情報を1つの状態とみなし、状態の同定および状態の接続の意味解析を行うことで、時相論理を用いた検査を行うことができると考えられる。

今後の課題として、以下の3つが挙げられる。

- より網羅的な修正候補の検出
- 手法の有用性の調査、および改善
- 節間の接続に着目した分析

1つ目について説明する。本研究の提案手法で対象とした修正候補は、節を1つの状態とした時に他の節から検出される状態を複数組み合わせられた節のみである。そのため、前後を表す表現やシステム全体の中での時間を表す表現に着目した解析を行うことを行いたい。

2つ目について説明する。本研究の提案手法では、時相論理に関係する単語を辞書登録し解析を加えた。この方法は、対象とする文書に強く依存するため、字句・構文解析による手法の検討を行い手法の改善を行いたい。

3つ目について説明する。本研究の提案手法では、節中の情報を用いて解析を行った。節間の接続に着目することで、状態の同定および状態の接続の意味解析ができると考えている。

参考文献

- [1] Bond, F., Kuribayashi, T., Isahara, H., Kanzaki, K. and Uchimoto, K.: 日本語ソーラス辞書 WordNet, , 入手先 (<http://compling.hss.ntu.edu.sg/wnja/>) (参照 2020 年 1 月 30 日).
- [2] 位野木万里, 近藤公久: 要求仕様の一貫性検証支援ツールの提案と適用評価, *SEC journal*, Vol. 13, No. 1, pp. 16–23 (2017).
- [3] 位野木万里, 近藤公久: 省略と修飾パターンを用いた用語不一致検証による要求仕様の一貫性検証支援ツールの実現と適用評価, *コンピュータソフトウェア*, Vol. 35, No. 3, pp. 3.109–3.127 (2018).
- [4] Li, W., Hayes, Jane, H. and Truszczyński, M.: Towards More Efficient Requirements Formalization: A Study., *Proc. of REFSQ 2015*, pp. 181–197 (2015).
- [5] 中村 成: 組込みシステムの要求仕様書を対象とした状態遷移表作成支援, 修士論文, 名古屋大学大学院情報学研究科 (2019). <https://sites.google.com/site/yoshidaatnu/naru.pdf>.
- [6] 中村 成, 山本椋太, 吉田則裕, 高田広章: 組込みシステムを対象とした要求仕様書からの状態遷移記述の抽出, *情報処理学会研究報告*, Vol. 2018, No. 5, pp. 1–8 (2018).
- [7] 大森洋一, 荒木啓二郎: 自然言語による仕様記述の形式モデルへの変換を利用した品質向上に向けて, *情報処理学会論文誌プログラミング (PRO)*, Vol. 3, No. 5, pp. 18–28 (2010).
- [8] 竹内広宜, 荻野紫穂, 中田武男, 坂本佳史, 福岡直明: テキスト分析技術を用いた開発関連文書の品質分析, *組込みシステムシンポジウム 2009*, pp. 93–100 (2009).
- [9] Wang, J.: *Real-Time Embedded Systems (Quantitative Software Engineering Series)*, Wiley (2017).
- [10] Wenbin, L.: Consistency Checking of Natural Language Temporal Requirements using Answer-Set Programming, PhD Thesis, College of Engineering at the University of Kentucky (2015). https://uknowledge.uky.edu/cs_etds/34.
- [11] 山本椋太, 中村 成, 吉田則裕, 高田広章: 要求仕様書を対象とした状態遷移記述抽出のための節分類手法の定量的評価, *日本ソフトウェア学会第 26 回ソフトウェア工学の基礎ワークショップ (FOSE 2019)*, pp. 57–62 (2019).
- [12] 山本椋太, 中村 成, 吉田則裕, 高田広章: 組込みシステムの要求仕様書に対する誤りと曖昧さの実証的調査, *電子情報通信学会論文誌 D*, Vol. J103-D, No. 5 (オンライン), 入手先 (https://search.ieice.org/bin/pdf_advpub.php?category=D&lang=J&fname=2019JDP7092&abst=) (2020).