Vol.2018-SE-198 No.5
2018/3/9

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

中村 成[1,a)]　山本 椋太[1]　吉田 則裕[1]　高田 広章[1]

概要：システム開発における要求仕様書は多くの場合，自然言語で記述される．そのため，目視による確認のみでは発見できない抜けや漏れが存在しうる．この問題を軽減するために，状態遷移モデルが考えられている．しかし，膨大な量のドキュメントから状態遷移モデルを手作業で作成する場合，開発者は状態遷移モデル作成に必要な状態遷移記述（状態名・状態値・イベント・状態値の判定・遷移・処理）を手作業で抽出する必要があるため，多大な時間を要する．そこで，本稿では誤字脱字や表現のゆらぎなどの日本語の誤りと曖昧さが除去されている要求仕様 1 文ずつに対して，状態遷移記述を含む要素の抽出・分類手法を説明する．提案手法は要求仕様 1 文ずつを入力として，状態遷移記述を含む要素とその分類結果を出力する．まず，状態遷移記述を含む要素や状態遷移記述を容易に抽出するために必要な要件・仕様を考察した．次に，要件・仕様を満たす解析木を導出し，解析木から状態遷移記述を含む要素の抽出を行った．その後，ツールにより抽出した状態遷移記述を含む要素に対して，分類条件を適用することで自動分類を試みた．その結果，要求仕様 19 文のうち，箇条書きの条件文 6 文に対しては手作業で抽出した場合と一致しなかったが，1 文で完結する文 13 文に対しては手作業で抽出した要素と一致したため 68%程度の精度を持つことが確認できた．

An Extraction of State Transition Descrptions in a Requirements Specification Document for an Embedded System

Naru Nakamura[1,a)]　Ryota Yamamoto[1]　Norihiro Yoshida[1]　Hiroaki Takada[1]

1. はじめに

システム開発における要求仕様書は多くの場合，自然言語で記述される．そのため，目視による確認のみでは発見できない抜けや漏れが存在しうる．また，組込みシステムの大規模化・複雑化によりソフトウェア開発期間の短縮と品質向上が求められる [1]．抜けや漏れの問題を軽減し，開発の上流工程において動作検証を行う状態遷移モデルが考えられている [2]．しかし，膨大な量のドキュメントから状態遷移モデルを手作業で作成する場合，開発者は状態遷移モデル作成に必要な状態遷移記述（状態名・状態値・イベント・状態値の判定・遷移・処理）を手作業で抽出する必要があるため，多大な時間を要する．

[1] 名古屋大学 Nagoya Uniersity
[a)] naru@ertl.jp

我々の研究グループでは，図 1 に示すように組込みシステムの要求仕様書から状態遷移モデル作成に必要な状態遷移記述の抽出を支援する手法の実現を目指している．

そこで，本稿では状態遷移記述を抽出するためのベースとなる，図 1 の破線の四角で囲まれた節の抽出手法を提案する．本研究における節とは，状態遷移記述を含み主語と述語が 1 つしか存在しない文または 1 文を分断した時に主語と述語が 1 つしか存在しない状態遷移記述を含む要素を指す．提案手法は，要求仕様 1 文ずつを入力とし，(a) から (g) までのステップを行うことで抽出した節とその分類結果を出力する．

(a) 要求仕様 1 文ずつを入力として，前処理（全角文字を半角文字へ変換，単位の表現を統一）を行う．
(b) 形態素解析システム JUMAN[3] で解析するために変換した要求仕様に対して JUMAN で形態素解析する．

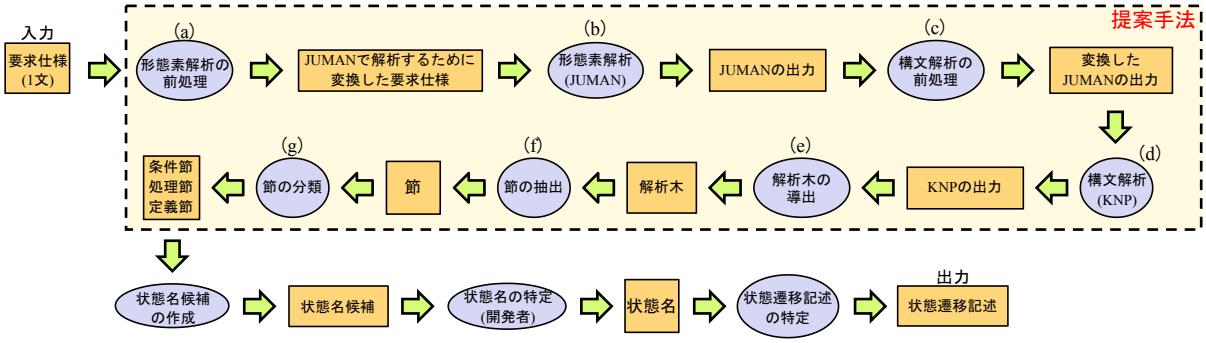

図 1 本研究の最終目標

(c) JUMAN の出力に対して構文解析の前処理（未定義語を名詞に置換，連続するべき名詞の結合，1 文を節単位で分割）を行う．
(d) 変換した JUMAN の出力に対して日本語構文・格・照応解析システム KNP[4] で構文解析する．
(e) KNP の出力に対して処理（判定詞と形容詞性述語接尾辞の結合，句読点と節の結合，修飾語と名詞の結合）を行い，解析木を導出する．
(f) 導出した解析木を用いて節の抽出を行う．
(g) 抽出した節に対して分類条件を適用することで節を分類する．

本稿では，実務者によりレビューによる修正がすでに加えられた要求仕様に対して，提案手法を実装したツールを適用する．その後，ツールが自動抽出した節の分類結果と第 1 著者が手作業で抽出した節を比較して，ツールの正確さを確かめる．

2. 利用した自然言語処理技術

自然言語処理には形態素解析，構文解析などがある [5]．形態素解析とは，意味を持つ最小の言語単位である形態素を解析するプロセスである．形態素解析の大まかな処理は入力文を形態素の並びに分割，形態素への品詞の付与，および形態素の原形の復元である．構文解析とは，文法として記述された言語の規則に基づいて文の構造を解析するプロセスである．構文解析の大まかな処理は各形態素を文法規則にしたがって結合，文節間の係り受けを決定，構文木の出力である．

2.1 形態素解析システム JUMAN

形態素解析システム JUMAN[*1]は，京都大学の黒橋・河原研究室で開発されたシステムである．入力として文字列を受け付け，辞書引きにより形態素ごとに分割し，その直前の位置に存在する全ての形態素との連接可能性のチェックおよびコストの計算を行う．この処理は入力文字列の終了を表す改行まで行われる．

出力される形態素情報には見出し，読み，原形，形態品詞，品詞細分類，活用型，活用形，代表表記などがある．形態品詞は日本語文法における品詞を表しており，品詞細分類は形態品詞をさらに分類した情報である．動詞のように，後接する形態品詞に応じて生じる形態変化を活用といい，大部分の活用は規則的である．その規則性に従って活用のタイプを分類したものを活用型，連接に応じて実際にとり得る表層的な個々の形態を活用形と呼ぶ．代表表記は，形態素のカテゴリやドメイン，意味情報などを指す．

2.2 日本語構文・格・照応解析システム KNP

日本語構文・格・照応解析システム KNP[*2]は，JUMAN と同様に京都大学の黒橋・河原研究室で開発されたシステムである．入力として JUMAN の出力を受け付け，形態素を文節および基本句単位にまとめて，それらの係り受け関係を決定する．末尾の形態素の係り受け番号は “-1” と表現される．しかし，現在の KNP では，文節内の基本句間の係り受けは扱っておらず，隣に係るように解析される．

KNP は，基本句単位で解析される．基本句とは，京都大学テキストコーパスに含まれる格関係，照応関係などの各種関係を付与する単位である．また，基本句は自立語一語を核として，その前後に存在する接頭辞，接尾辞，助詞，助動詞などの付属語で構成情報されており，文節と同じ，もしくは一部である．

3. 分析対象の要求仕様書

3.1 使用する要求仕様書の概要と修正

本研究では，企業において実際に使用されている車載組込みシステムに関する要求仕様書（以降 D_0 と呼ぶ）を用いる．本研究で使用する要求仕様書は，誤字脱字や表現のゆらぎなどの日本語の誤りと曖昧さが除去されていることを前提とする．そこで，本稿では，実務者が D_0 の 19 文に対してレビューによる修正を加えた要求仕様（以下，厳密な要求仕様）を使用する．厳密な要求仕様とは，以下の 3.1.1 から 3.1.5 の各節で示す処理を手作業で行ったもので

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

ある．

3.1.1 表現ゆれの修正

本来，統一されるべき箇所の表現が統一されていないため，修正する．以下の例の修正前においては，「LED が『オフ』である」という表現と，「『オン』が 50[ms] 以上継続しないで」という異なる表現が存在するため，修正後には接続詞を補完し，「LED が『オン』である」という表現に統一した．

修正前）LED が「オフ」であるか、「オン」が 50[ms] 以上継続しないで「オフ」に変化した。

修正後）LED が「オフ」である、または LED が「オン」である状態が 50[ms] 以上継続していない状態から「オフ」に変化した。

3.1.2 動作の状態化

本来，状態で表現されるべき箇所が動作として表現されているため，修正する． 以下の例の修正前においては，「LED が『オン』である状態が 50[ms] 以上継続した後に」という表現により，動作として表現されているが，修正後には「状態」を補完し，「LED が『オン』である状態が 50[ms] 以上継続した状態」という状態に修正した．

修正前）LED が「オン」である状態が 50[ms] 以上継続した後に「オフ」に変化した場合、LED 押下判定を「オン」にする。

修正後）LED が「オン」である状態が 50[ms] 以上継続した状態から「オフ」に変化した場合、LED 押下判定を「オン」にする。

3.1.3 箇条書きの条件文の前後にある文の記述位置を統一

箇条書きの条件文の前後にある文の位置が箇条書きの前後どちらかに統一されていないため，箇条書き以外の文を箇条書きの前に統一する．以下の例の修正前においては，「いずれかが成立していない場合は許可しない」が箇条書きの直後にあるため，修正後には「それ以外の場合、入力を不許可にする。」のように文を修正し，箇条書きの直前に記述した．

修正前）以下の条件が全て成立している場合、入力を許可する。

・スイッチ A が「オン」である

・診断の結果、スイッチ B に異常が発生していない

いずれかが成立していない場合は許可しない。

修正後）以下の全ての条件が成立している場合、入力を許可する。それ以外の場合、入力を不許可にする。

・スイッチ A が「オン」である

・診断の結果、スイッチ B に異常が発生していない

3.1.4 用語の統一

同一の意味を持つ語が異なる表記になっているため，修正する．以下の例の修正前においては「許可しない」という語は修正後のように「不許可にする」という語に修正する．

修正前）入力を許可しない。

修正後）入力を不許可にする。

3.1.5 主語の明確化

文中の主語を明確にするため，修正する．以下の修正例においては「として」では主語がないため，修正後のように「は」という直前の要素が主語とわかる語に修正する．

修正前）スイッチ A の状態として、「オン」と「オフ」の状態を保持する。

修正後）スイッチ A の状態は「オン」「オフ」のいずれかの値を取る。

3.2 構文木の導出

要求仕様 1 文に対して，まず JUMAN を用いて形態素解析する．次に．JUMAN の出力に対して KNP を用いて構文解析する．

その後，構文解析の結果から構文木を作成する．

例文 1) スイッチが「オン」である時にスイッチ押下判定が「オン」に変化した場合、ＬＥＤを「オフ」にする。

例文 1 に対して，形態素解析，構文解析を行い得た構文解析の結果から導出した構文木を**図 2** に示す．ただし，基本句を作成する形態素の結合は省略する．

4. 提案手法

本稿では，組込みシステムの要求仕様 1 文ずつに対して節の抽出・分類手法を提案する．1 章で述べた通り，節とは，状態遷移記述を含み主語と述語が 1 つしか存在しない文または 1 文を分断した時に主語と述語が 1 つしか存在しない状態遷移記述を含む要素を指す．しかし，構文木には連続する名詞を結合して得られる要素や，1 文を構成する全ての節が存在しないなどの問題がある．図 2 の構文木を用いると，「スイッチ押下判定が『オン』に変化した場合、」と「LED を『オフ』にする」という 2 つの節と，「スイッチ押下判定」のように連続する名詞を表す要素が構文木に存在しないことが分かる．そこで，これらの問題を解決するために本研究では，構文木を使用せず，節や状態遷移記述を容易に抽出できる解析木を新たに導出し，使用する．解析木の要求を以下に示す．

- 要求 1：全ての形態素が形態品詞に未定義語を持たないこと．
- 要求 2：連続するべき名詞が結合すること．
- 要求 3：文を構成する節が全て存在すること．
- 要求 4：判定詞と形容詞性述語接尾辞が結合していること．
- 要求 5：節の末尾に句読点がある場合，句読点が最後に結合すること．

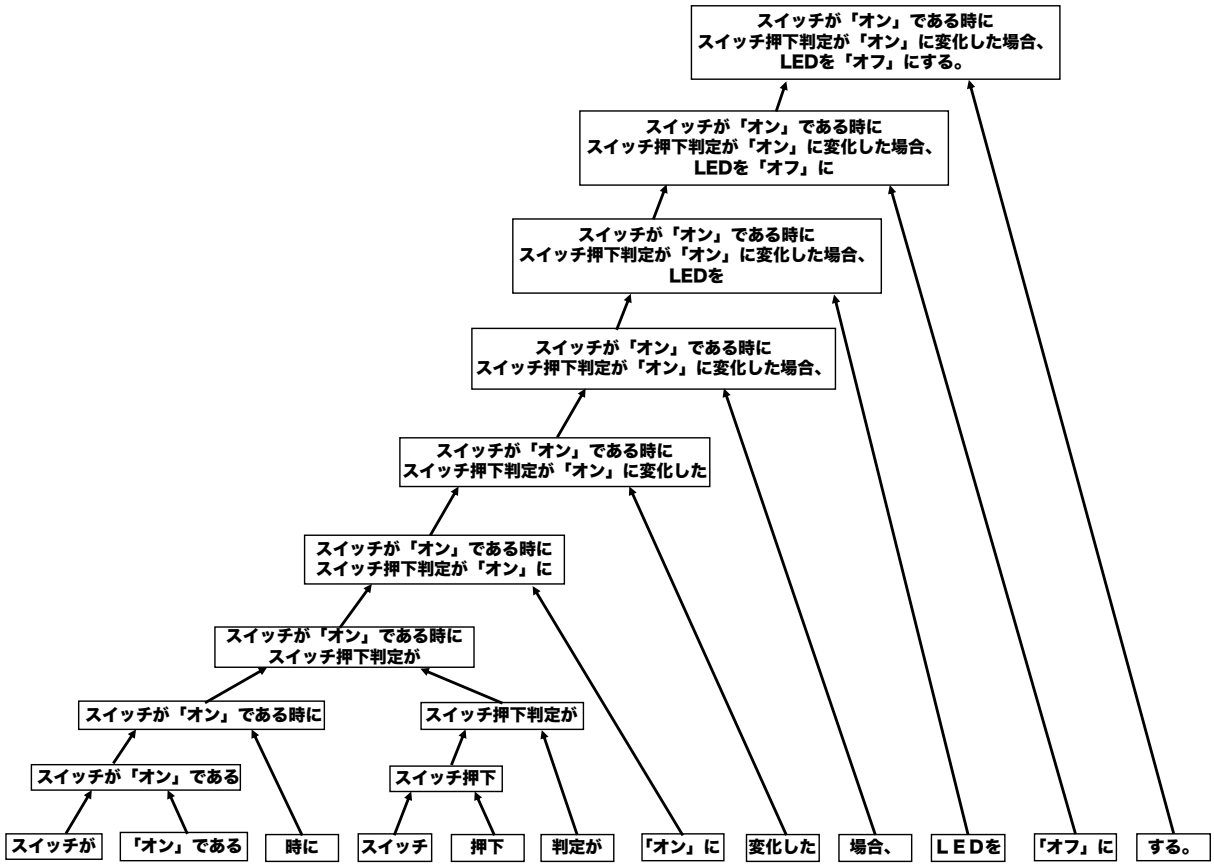

図 2 例文 1 の構文木

表 1 解析木の仕様

	仕様	例
1	形態品詞に未定義語を持つ形態素を全て名詞に置換する（KNP を用いた構文解析の前処理）	「インジケータセル」,「メトリクス」など
2	（名詞 +（名詞|接尾辞））が存在する時，結合する（KNP を用いた構文解析の前処理）	「スイッチ + 押下 + 判定」,「モデル + 化」など
3	副詞的名詞 +（格助詞|接続助詞|副助詞|読点 ））|句点|（判定詞 + 読点）が存在する時，その形態素列で JUMAN の出力結果を分断する（KNP を用いた構文解析の前処理）	「時 + に」,「時 + の」,「場合 + は」,「場合 + 、」,「である + 、」など
4	判定詞 + 形容詞性述語接尾辞が存在する時，結合する（KNP の結果から作成した部分木の修正）	「で + ない」
5	（副詞的名詞|動詞|判定詞）+（句点|読点）が存在する時，分断する（KNP の結果から作成した部分木の修正）	「する + 。」,「場合 + 、」など
6	（⟨ 接続助詞 ⟩+ ⟨ 副詞的名詞以外の名詞 ⟩）| （⟨ 接続助詞 ⟩ + ⟨ 形容詞 ⟩）| ⟨ 動詞性接尾辞 ⟩ +⟨ 副詞的名詞以外の名詞 ⟩）| （⟨ 形容詞性述語接尾辞 ⟩ + ⟨ 副詞的名詞以外の名詞 ⟩）| （⟨ 判定詞 ⟩ + ⟨ 副詞的名詞以外の名詞 ⟩）| （⟨ 動詞 ⟩ + ⟨ 副詞的名詞以外の名詞 ⟩）が存在する時，結合する（KNP の結果から作成した部分木の修正）	LED「の + 信号」，LED「の + 異常」，学習して「いない + 状態」，考え「られる + 状態」，静か「である + 状態」，加速「した + 状態」など

- 要求 6：修飾語と名詞が結合すること．

要求 1 から 6 を満たすために策定した解析木の仕様 1 から 6 を**表 1** にまとめる．

本章では，**図 1** に示すように破線の四角で囲まれた提案手法について説明する．提案手法の流れは，1 章および図 1 で示した（a）から（g）である．

4.1 形態素解析の前処理

JUMAN を用いた形態素解析を行う前に（a-1），(a-2）を行う．

（a-1）：入力文の半角文字を全て全角文字に変換する．

（a-2）：アルファベットの単位を日本語に変換する．

JUMAN は，全角文字列に対する処理システムであり，半角文字列の辞書登録，解析をサポートしていない．そのため，要求仕様に半角文字が含まれていると，正確な解析結果が得られない．したがって，(a-1）を行う．

要求仕様には，時間や距離を表すために [ms] や [km] などの単位が用いられる場合がある．この単位を JUMAN で形態素解析すると，ms や km などのアルファベットで記述される単位は未定義語に分類され，他の語との結びつきが不適切になる．しかし，ms や km をミリ秒やキロメートルなどの日本語に変換すると，未定義語ではなく，適切な形態品詞，品詞細分類に分類される．したがって，“[ ]” で囲まれている単位ごと（a-2）を行う．

4.2 JUMAN の出力の変換

表 1 にまとめた解析木の仕様のうち，仕様 1，仕様 2，および仕様 3 は，JUMAN の出力を変換することで実現する．まず，仕様 1 について説明する．JUMAN で形態素解析を行うと，要求仕様書の記述に含まれる専門用語を未定義語として分類した．未定義語と分類された形態素のうち，名詞とみなすことができる形態素を目視で確認することができた．この形態素の前後に，名詞が存在する時，要求 1 を満たすために結合を試みるが，未定義語と分類されているため，（名詞 + 名詞）の規則を満たすことができない．そのため，形態品詞が未定義語である形態素を全て名詞に置換する．今回は，半角文字だと未定義語だが，全角文字に変換すると名詞として扱われる形態素「ON」の，全角文字の場合の形態素情報をベースとした．現在，名詞に置換したことによる問題は発生していない．したがって，要求 1 を満たすことができる．

次に，仕様 2 について説明する．まず，JUMAN の出力を出力結果順にたどり，隣り合う形態素の形態品詞が名詞の場合，2 つの連続する形態素を結合する．結合して得られる情報のうち，表記，読み，原形に関しては 2 つの形態素情報を結合し，これら 3 つの情報と代表表記以外は 2 つ目の形態素情報を継承する．次に，結合するために用いた 2 つの形態素は，JUMAN の出力から削除する．その後，結合して得られる要素を始点に更新し，隣り合う形態素の形態品詞が名詞以外になるまで，この手順を繰り返す．また，形態品詞に名詞を持つ形態素と形態品詞に接尾辞を持つ形態素が隣り合う場合は，結合して得られる情報のうち表記，読み，原形以外の情報に対して 1 つ目の形態素の情報を継承することを除き，同様の手順で結合する．この処理により，要求 2 を満たすことができる．

表 2 例文 1 に対して仕様 2，仕様 3 を実現した結果

節番号	節内容
節 1	スイッチ が 「 オン 」 である 時 に
節 2	スイッチ押下判定 が 「 オン 」 に 変化 した 場合 、
節 3	ＬＥＤ を 「 オフ 」 に する 。

次に仕様 3 について説明する．仕様 3 は，JUMAN の出力に仕様 3 を満たす規則のいずれかを持つ場合，その形態素の並びで 1 文を区切る．要求仕様書に目を通すと，仕様 3 の規則を持つ形態素を区切りの形態素であると考えた．このとき，区切りの形態素毎に，区切りの形態素を終端形態素として結合すると節を形成できることが分かった．よって，節になりうる形態素ごとに分割した JUMAN の出力に対して KNP を用いた構文解析を行うと，部分木の根が必ず節となるため，要求 3 を満たすことができる．例文 1 に対して仕様 2，仕様 3 を実現した結果を**表 2** にまとめる．ただし，見出し以外の形態素情報は省略する．

4.3 解析木の導出

表 1 に示した仕様をツールに実装する．仕様 1，仕様 2，および仕様 3 は，KNP を用いた構文解析の前処理で実現する．仕様 4，仕様 5，および仕様 6 は，構文解析の結果に対して実現する．この処理を行うと，最終的に，要求 1 から要求 6 を満たす解析木が導出される．

解析木の導出手順は，まず要求 1 と要求 2，および要求 3 を満たすために 4.2 節で述べた通り JUMAN の出力を変換する．次に，変換した JUMAN の出力に対して KNP を用いた構文解析を行う．さらに，構文解析の結果に対して，以下の手順（e-1）から手順（e-7）を行い，解析木を導出する．

（e-1）文頭から順に各形態素を基本句単位で結合する．

（e-2）文節内に基本句が複数ある場合は，それらを結合して文節を得る．

（e-3）先頭から順に，係り受け先の文節と結合する．結合して得た要素は次に結合すべき文節として，自身の結合に用いた文節のうち，文節番号の大きい文節が持つ係り受け先の文節番号を継承する．

（e-4）同じ係り受け先の文節番号を持つ文節が存在する場合は，それらを先に結合する．

（e-5）（e-3）と（e-4）を繰り返して部分木を導出する．

（e-6）仕様 4，仕様 5，および仕様 6 を実現する．

（e-7）各部分木の根ノードを部分木の導出順に結合する．

表 3 の情報に対して（e-1）から（e-7）の手順を行い，導出した解析木を**図 3** に示す．

4.4 状態遷移記述の抽出

図 3 の解析木を見ると，「スイッチ押下判定」が 1 つの名詞として見なされているため，要求 2 を満たす．また，例

表 3 解析木導出に用いる情報

文節番号	文節	基本句	係り受け先の文節番号	節
1-0	スイッチが	スイッチが	1-1	
1-1	「オン」である	「 オン 」である	1-2	節 1
1-2	時に	時に	-1	
2-0	スイッチ押下判定が	スイッチ押下判定が	2-2	
2-1	「オン」に	「オン」に	2-2	節 2
2-2	変化した	変化した	2-3	
2-3	場合、	場合、	-1	
3-0	ＬＥＤを	ＬＥＤを	3-2	
3-1	「オフ」に	「オフ」に	3-2	節 3
3-2	する。	する。	-1	

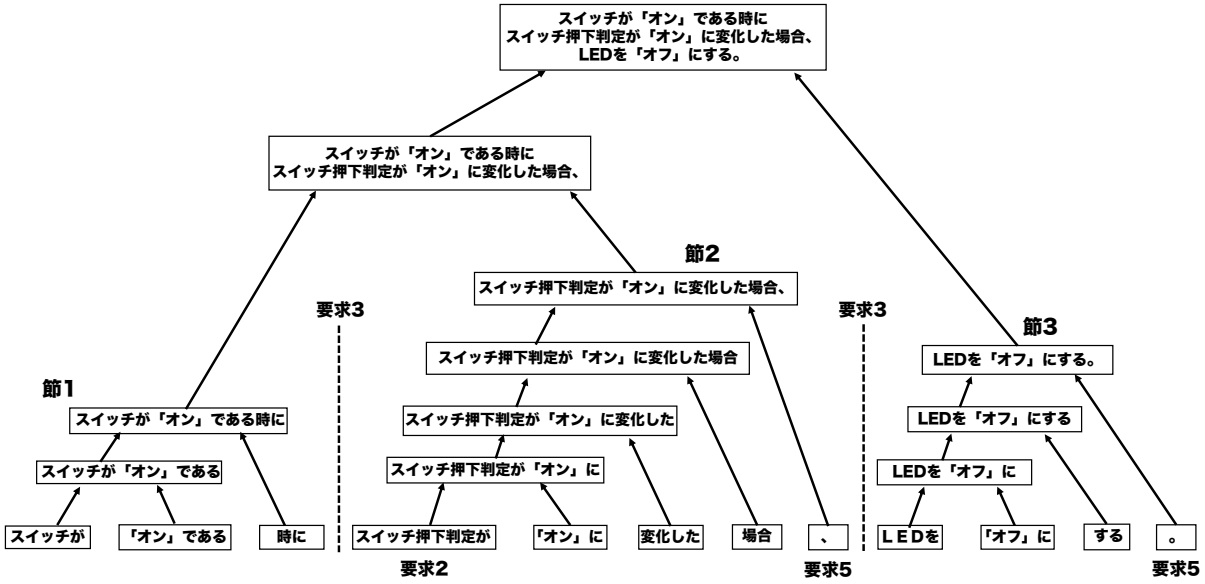

図 3 例文 1 の解析木

文 1 は節単位で区切られ，例文 1 を構成する「スイッチが『オン』である時に」，「スイッチ押下判定が『オン』に変化した場合、」，「ＬＥＤを『オフ』にする。」といった 3 つの節が解析木の要素として全て存在するため，要求 3 を満たす．さらに，句読点を持つ節「スイッチ押下判定が『オン』に変化した場合、」と「ＬＥＤを『オフ』にする。」は最後に句読点と結合するため要求 5 を満たす．したがって，解析木は全ての要求を満たすため，解析木を用いて，1 文に存在する節を全て抽出することが可能になる．次に，抽出した節の分類を試みる．

分類することを考えている本研究における節の名称と定義を以下に示す．

- 条件節：条件を表す節
- 処理節：条件を満足した時に起こる節
- 定義節：状態を定義している節

これら 3 つの節を抽出する理由は，各節に特徴があるため分類が容易であり，次に抽出する予定の状態名候補を見つける際に使うためである．本研究で考察した節の分類条件と例を**表 4** にまとめる．

5. ケーススタディ

本章では，提案手法を実装したツールで分類した節と，第 1 著者が手作業によって抽出した節を比較する．本事例の対象は，3 章で述べた厳密な要求仕様 19 文である．

節の分類例として，3.2 節で示した例文 1 と以下に示す例文 2 の分類結果を**表 5** にまとめる．

例文 2) LED は「オン」「オフ」のいずれかの値を取る。

例文 1 の「スイッチが『オン』である時に」と「スイッチ押下判定が『オン』に変化した場合、」は条件節，「ＬＥＤを『オフ』にする。」は処理節，例文 2 の「ＬＥＤは『オン』『オフ』のいずれかの値を取る。」は定義節に分類された．

この要求仕様書には，箇条書きの条件文が存在する．箇条書きの条件文とは，「以下の全ての条件が成立している場合、」などの節を含む文がある時，「以下の」を示すため

表 4 節の分類条件

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。

表 5 本研究における節の分類例

条件節	スイッチが「オン」である時に スイッチ押下判定が「オン」に変化した場合、
処理節	ＬＥＤを「オフ」にする。
定義節	ＬＥＤは「オン」「オフ」のいずれかの値を取る。

表 6 箇条書きの条件文の分類例

定義節	ＬＥＤが「オン」である。
定義節	入力が許可である。

に直下に記述される詳細な条件文である．箇条書きの条件文の例を以下に示す．

- 箇条書きの条件文 1：LED が「オン」である。
- 箇条書きの条件文 2：入力が許可である。

箇条書きの条件文 1，2 の分類結果を**表 6** にまとめる．その結果，箇条書きの条件文は 2 文とも定義節に分類された．

厳密な要求仕様 19 文に対して 13 文は，正しい分類と判断した．この 13 文は箇条書きの条件文以外の文であり，例文 1，例文 2 のように 1 文で完結する文である．13 文の分類が正しいと判断した理由は，手作業で抽出した節と一致していることが確認できたためである．

残りの 6 文は箇条書きの条件文であり，全て誤った分類と判断した．この 6 文が誤りと判断した理由は，箇条書きの条件文は直前に記述される箇条書き以外の文に含まれる「以下の」に対応する条件文であり，本来は条件節と分類されるためである．誤った分類を行う原因には，以下の理由が考えられる．要求仕様書は Excel で記述されており，「以下の」を含む文と箇条書きの条件文は同じセル内に記述されている．しかし，ツールは，セル内の要求仕様 1 文ずつに対して提案手法を適用する仕様である．そのため．同じセル内の文であることを対応させることができず，1 文に対して分類条件を適用するため，定義節と分類された．

本事例より，ツールは厳密な要求仕様 19 文に対して，約 68%の精度で分類することができることが確認できた．現在，箇条書きの条件文の分類は，実装を進めている．

6. 関連研究

本研究の前提として，要求仕様書の日本語の誤りと曖昧さを除去する必要がある．そのための手法は，数多くの研究が存在する．まず，山本ら [6] は，組込みシステムのソフトウェアに関する要求仕様書を目視により確認して，出現した誤りや曖昧さを項目ごとに分類している．分類した修正候補の項目のうち，実装が容易であったものを優先して実装した結果，目視による確認と比べて精度の高い抽出が行えている．したがって，分類した修正候補を全て自動抽出することができれば，要求仕様書の誤りと曖昧さは減少することが考えられる．次に，今枝ら [7] や南保ら [8] のように，NTT 日本語語彙体系 [9] のような辞書を用いて，書き誤る可能性が高い日本語助詞の検出・修正を行う手法が挙げられる．Mizumoto ら [10] は，相互添削型言語学習 SNS である Lang-8 の添削履歴を用いて，日本語の誤り訂正を行う技術を提案している．笠原ら [11] は，日本語学習者の格助詞誤り傾向を反映して， 格助詞訂正手法を提案している．

Desai ら [12] は自然言語とそれに対応するドメイン固有言語（DSL）の組と DSL の定義をトレーニングデータとして機械学習させることで，自然言語から DSL を生成するフレームワークを構築している．本研究においても，自然言語とモデルの組を学習させることで，状態遷移記述を自動抽出できる可能性が考えられる．

山本ら [13] は，ソースコードから状態遷移表を抽出する研究を行っているため，要求仕様書とソースコードが一致しているかどうかを確認することができると考えられる．

7. おわりに

本稿では，組込みシステムの要求仕様書から節の抽出・分類手法を提案した．提案手法を実現するために，形態素解析の前処理，形態素解析，構文解析の前処理，構文解析，解析木の導出，節の抽出，節の分類の 7 つの処理を行った．節や状態遷移記述を容易に抽出できる解析木を導出するために満たすべき 6 つの要求と仕様を考察した．その後，解析木を用いると，節を自動抽出できることが確認できた．

次いで，ツールで節を自動分類し，第 1 著者が手作業によって抽出した節と比較した．その結果，要求仕様 19 文に対して，約 68%の精度で分類することができることが確認できた．

今後の課題は，箇条書きの条件文を分類できるようにツールを改良し，必要な要素をさらに抜き出す抽出支援システムの開発を予定している．また，提案手法を他の要求仕様書に対して適用することも考えている．

参考文献

[1] 高田広章：組込みシステム開発技術の現状と展望，情報処理学会論文誌， Vol. 42, No. 4, pp. 930–938 (2001).
[2] 渡辺政彦：状態遷移ベースのソフトウェア開発環境の現状と動向，計測と制御，Vol. 41, No. 2, pp. 117–121 (2002).
[3] 黒橋禎夫：日本語形態素解析システム JUMAN version 7.0 使用説明書 (2012).
[4] 黒橋禎夫：日本語構文解析システム KNP version 4.1 使用説明書 (2013).
[5] 黒橋禎夫：自然言語処理，放送大学教育振興会 (2015).
[6] 山本椋太，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修正候補の定量的調査，電子情報通信学会技術研究報告，Vol. 117, No. 249, pp. 49–54 (2017).
[7] 今枝恒治，河合敦夫，石川裕司，永田 亮，桝井文人：日本語学習者の作文における格助詞の誤り検出と訂正，Vol. 2003, No. 13 (2002-CE-068), pp. 39–46 (2003).
[8] 南保亮太，乙武北斗，荒木健治：文節内の特徴を用いた日本語助詞誤りの自動検出・校正，情報処理学会研究報告自然言語処理 (NL)， Vol. 2007, No. 94 (2007-NL-181), pp. 107–112 (2007).
[9] 池原 悟，宮崎正弘，白井 諭，横尾昭男，中岩浩巳，小倉健太郎，大山芳史，林 良彦：日本語語彙大系，岩波書店 (1997).
[10] Mizumoto, T., Komachi, M., Nagata, M. and Matsumoto, Y.: Mining Revision Log of Language Learning SNS for Automated Japanese Error Correction of Second Language Learners., *Proc. of IJCNLP 2011*, pp. 147–155 (2011).
[11] 笠原誠司，藤野拓也，小町 守，永田昌明，松本裕治：日本語学習者の誤り傾向を反映した格助詞訂正，言語処理学会第 18 回年次大会，pp. 14–17 (2012).
[12] Desai, A., Gulwani, S., Hingorani, V., Jain, N., Karkare, A., Marron, M., Roy, S. et al.: Program synthesis using natural language, *Proceedings of the 38th International Conference on Software Engineering, Proc. of ICSE 2016*, ACM, pp. 345–356 (2016).
[13] 山本椋太，吉田則裕，青木奈央，高田広章：組込みソフトウェアを対象とした状態遷移表の抽出と分析支援の検討，電子情報通信学会技術研究報告，Vol. 117, No. 136, pp. 133–138 (2017).