

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

An Empirical Study of Clause Categorization in Extracting State Transition Descriptions of a Requirements Specification Document

山本 椋太* 中村 成† 吉田 則裕‡ 高田 広章§

あらまし 一般にシステム開発における要求仕様書は自然言語で記述され、発見が困難な抜けや漏れが存在し、この問題を軽減するため、状態遷移モデルが広く利用されている。組込みシステムの要求仕様書からの状態遷移モデル作成支援ツールの開発を行った先行研究があり、そのツールに含まれる手法の1つに節分類手法があるが、定量的評価が十分ではない。本稿では、先行研究にて提案された節の分類手法を定量的に評価する。その結果、本稿にて対象にした要求仕様書は高い精度に節を分類できたが、新たに「制約節」の追加を検討する必要があるとわかった。

1 はじめに

システム開発においては、様々な文書が自然言語によって作成される [1]。自然言語による記述は、認識の相違や、目視によって発見することが困難な抜けや漏れにつながりうる。この問題を軽減するべく、ソフトウェア開発の上流工程において、状態遷移モデルが広く利用されている [2]。しかし、要求仕様書は2,000 ページ程度になることもあり [3]、開発者が自然言語の文書をすべて読み、状態遷移モデルの作成に必要な要素をすべて正しく抽出することは、非常に高コストかつ高難度である。

この問題を軽減するべく、我々の研究グループでは自然言語によって記述された組込みシステムの要求仕様書から、状態遷移モデルを抽出する支援手法の検討を進めてきた（以降、先行研究と呼ぶ） [4] [5]。先行研究の手法では、自然言語によって記述された組込みシステムの要求仕様書から独自の解析木を導出して節を抽出し、定義節、条件節および処理節のいずれかに分類する。分類された節を利用して、状態遷移記述の抽出を試みる。これらの手法を実装したツール（以下、解析ツールと呼ぶ）が存在しており、特定の使用条件下においては状態遷移モデルを抽出可能であることがケーススタディによって示されている [5]。先行研究における貢献の1つが節の分類であり、解析ツールは、ユーザによるインタラクティブな操作に基づいて、分類された節から状態遷移モデルにおける状態、イベント、処理および遷移を抽出する。しかし、節の分類精度について十分な評価が行われておらず、先行研究における提案手法が、どの程度節を正しく分類可能か不明である。そこで、本稿では以下のリサーチクエスチョン（RQ）を設定した。

RQ1: 節の分類精度はどの程度か

RQ2: 先行研究の提案手法では、分類できない節の種別が存在するか

本稿では、以上の RQ を調査するための実験を実施し、解析ツールによって解析可能な自然言語の文を調査したため報告する。

2 実験

本稿では、1の2つの RQ を調査するべく、以下の2つの実験を実施した。

*Ryota Yamamoto, 名古屋大学

†Naru Nakamura, 名古屋大学

‡Norihito Yoshida, 名古屋大学

§Hiroaki Takada, 名古屋大学

実験 1 節の分類の precision および recall の調査

実験 2 箇条書きの分類の precision および recall の調査

本実験においては、先行研究にて用いられた、修正済みの車載システムの要求仕様書を使用した。修正内容については、2.1 にて述べる。実験実施者は第 1 著者であり、あらかじめ第 1 著者が節の分類を手作業によって実施した。これらの修正内容は、解析ツールが修正を軽減できるように対応するべきであり、本稿で調査対象とする手法においては要求仕様書の修正量の多さについて議論しない。

ここで、解析ツールは箇条書きの文は対象とせず解析を実行している。その理由は、箇条書きの解析は、箇条書きを参照している親となる文中の、「以下の」をキーとしており、「以下の」が含まれている節の種別を継承するためである。そのため、別途**実験 2**を設け、箇条書きの各項目が正しい節種別となるかを確認した。

2.1 適用対象

本稿では、実際に企業にて使用された車載システムの要求仕様書を使用する。この要求仕様書は、もともと [5] におけるケーススタディのために用いられたもので、変更が加えられている。変更の結果、状態遷移記述に関する記述以外は削除されている。この要求仕様書は構造化されており、機能要求と非機能要件が分離している。また、機能要求の内容と、説明および理由が分離されている。[5] において使用された、機能要求の内容のみを本稿で利用する。この結果、文単位に分割されたテキストファイルの数は 243 となった。

2.2 実験手順

実験 1 は、以下の手順で実施された。

手順 1 先行研究にて開発された解析ツールを実行する。

手順 2 2.3 で説明するインタラクティブな操作を第 1 著者が実施する。

手順 3 解析ツールが生成した解析木を確認し、第 1 著者が節の分類と比較する。

実験 2 は、上記**手順 3**から引き続き実施される。

手順 4 箇条書きの文と箇条書きを参照する文を、ツールの出力結果から目視で第 1 著者が確認する。

手順 5 箇条書きの文に対して、解析ツールがその箇条書きの分に対応する箇条書きを参照する文を分類した結果を割り当てる。

手順 6 **手順 5**の結果を第 1 著者による節の分類と比較する。

いずれの実験においても第 1 著者が正解集合を作成しているが、これらは解析ツールの結果を見ず、第 1 著者自身が手作業で分類した。

実験 1 および**実験 2**における比較では、適合率 (*precision*) および再現率 (*recall*)、これらの調和平均である F 値 (*F*) を算出する。正解集合を第 1 著者が分類した節の分類集合 **A** とし、解析ツールが分類した節の集合を **E** とする。解析ツールが分類した節の分類のうち正解集合に属する個数を正解数とするとき、 $precision = |\mathbf{A} \cap \mathbf{E}|/|\mathbf{E}|$ 、 $recall = |\mathbf{A} \cap \mathbf{E}|/|\mathbf{A}|$ と定義する。

2.3 実験結果

実験 1の結果を表 1 に示す。表 1 中の X は不定値を表し、括弧の外は第 1 著者によるインタラクティブな操作を含む抽出結果であり、括弧の中は解析ツールが自動抽出のみによって抽出した結果である。表 1 中の括弧の外の数値は解析ツールによる完全な自動抽出結果を、括弧の内の数値は第 1 著者によるインタラクティブな操作による結果を示している。インタラクティブな操作は、解析ツール上で、ユーザが「条件節と処理節の組み合わせ」であるか、それとも定義節であるかを判断する操作である [6]。今回、該当する文は 3 文あり、それらはすべて定義節と判断した。

また、本実験においては分類不明な節が存在していた。解析木の情報とは、文または文の一部である節の文字列と、その節の種別である。未分類とは、節の種別が

表 1 実験 1 の結果

対象	A	E	A ∩ E	<i>precision</i>	<i>recall</i>	<i>F</i>
条件節	112	113(113)	112(112)	0.991(0.991)	1.000(1.000)	0.996(0.996)
処理節	85	86(86)	85(85)	0.988(0.988)	1.000(1.000)	0.994(0.994)
定義節	33	32(29)	32(29)	1.000(1.000)	0.967(0.879)	0.985(0.935)
未分類	0	4	0	0	X	X
合計	230	235(232)	229(226)	0.974(0.974)	0.996(0.983)	0.985(0.978)

表 2 実験 2 の結果

対象	A	E	A ∩ E	<i>precision</i>	<i>recall</i>	<i>F</i>
条件節	195	191	191	1.000	0.979	0.990
処理節	7	3	3	1.000	0.429	0.600
定義節	0	0	0	X	X	X
合計	202	194	194	1.000	0.960	0.980

与えられていない節である。未分類は 4 つ存在しており、それらはすべて箇条書きの文であり、かつ条件節と処理節の組み合わせからなる節であった。

次いで、実験 2 について述べる。結果を表 2 に示す。表 2 中の X は、不定値を表す。実験 2 の結果は、インタラクティブな操作に対する依存はない。本実験においては、数式も箇条書きと分類されるが、解析ツールの制約上、数式は解析対象外であるため、今回の結果に含めていない。

3 考察

3.1 RQ1: 節の分類精度

実験 1 の結果から、箇条書きを除く文については、総じて非常に高い F 値を得ている。特に、インタラクティブな操作を含む場合では、F 値は 0.98 を超えている。また、自動抽出のみによる結果でも、F 値で見れば 0.9 を超えており、良い結果を得ることができていると言える。しかし、この結果から自動抽出だけでは、軽微ではあるが分類精度が低下するということがわかる。自動抽出によって判断できないものは、条件節と処理節の連続からなる文に見えるが、「制約の文」かどうか判断することが困難なものである。すなわち、条件節に伴う文が、振る舞いではなく性質や制約を表している場合があり、条件節に伴う文が振る舞いを表していればそれは処理節であり、そうでなければ定義節と分類されるべきである。この分類を行う手法が決定できれば、精度をさらに向上できる可能性がある。

次に、実験 2 の結果から、全体としての F 値は 0.9 を超えているが、処理節の精度が低い結果となった。これは、実験 1 で未分類となった箇条書きの文が解析されていないことによる。これら 4 つの文は、すべて条件節と処理節の組である。そのため、手作業による抽出結果の方が条件節と処理節が各 4 つずつ多くなった。箇条書きの中に複数の節がある場合の分類規則が不足であるため、これは手法が改良される必要がある。それ以外の箇条書きは、今回対象とした要求仕様書中のすべての箇条書きを参照する節中に「以下の」が含まれており、すべて正しく抽出できた。

文および節の分類精度は、手法によって定義されていない構文が存在し得た場合、精度が低下する恐れがある。今回の修正を加えた 243 文からなる要求仕様書に対しては、高い精度で節の分類ができた。

また、未分類となった節は、すべて実際には箇条書きであった。

以上から、RQ1 について、以下のようにまとめる。

- 自動抽出であっても 0.9 を超える F 値によって節種別を正しく分類できる。

- 箇条書きについては、特定のキーワードを正しい位置に含むことで分類できる。
- 箇条書き中に複数の節がある場合の解析方法の検討は、今後の課題である。

3.2 RQ2: 節の分類種別

まず、RQ1でも述べたとおり、実験1の結果における未分類は節の定義が不足していたわけではなく、手法としての今後の課題である。そのため、実験1、実験2において、節種別の不足はない。

実験1にて分類のためにインタラクティブな操作を必要とした節について考察する。先行研究における手法では、条件節、処理節および定義節の3種の節を定義している。条件節と処理節は基本的には組であり、定義節はそれだけで存在する。実験2によって示された通り、条件節や処理節も単独で存在する可能性はあり、それは、箇条書きの各項目である。条件節は、「条件」であるため、条件に伴う動作が必ず必要であるため、箇条書きの項目である場合を除けば、必ず後ろに従属する節が必要となる。3.1においても述べたが、条件節の後ろに来る節が処理節ではないことも考えられる。実際、実験1においては、第1著者は条件節と処理節の組であるとも考えることが可能な文の最初の節を「条件節ではない」と判断している。この結果、条件節となりうる箇所も含め、解析ツールによって文全体が定義節であると分類された。この結果から、箇条書きに関しては節の分類が十分だと判断している。

次に、インタラクティブな操作が必要な節について検討する。現状、副詞的名詞（「とき」、「際」、「場合」など）と接続助詞（「ので」、「けれど」など）の連続が節末尾に存在した場合に、条件節と処理節の組か判断ができず、ユーザによるインタラクティブな操作が必要である。書き方によって条件節に伴う処理、および条件節に伴う制約は相互に書き換え可能な場合がある。しかし、最終的に状態遷移モデルを抽出することを考えるならば、これらの表現の差は重要なものであると考える。つまり、制約であれば状態遷移モデルに含まない要素となるため、処理として扱われなければ、最終的にはイベントや処理・遷移の抜けとなる可能性がある。しかし、特定の場合だけに処理として実装されるのではなく、常に成り立たなければならない不変条件であるならば、それはイベントとして表現することは困難であり、処理ではなく制約となる。このように、書き手が、「処理」を意識して仕様書に記載したのか、「制約」を意識して記載したのかは、解析ツールが目的としている状態遷移モデルの抽出に対して影響を与える可能性がある。

次いで、定義節と制約の文の比較について考察する。定義節は、「状態を定義している節」と定義している。対して、制約の文は禁止状態だけを示すことが多く、型または変域の定義とは目的が異なり、定義節の目的と一致していないと考える。

現状、実験1においてすべての文を分類できたが、上記の問題があるため、定義節から制約節を分離することが望ましいと考えている。

以上から、RQ2を以下のようにまとめる。

- 現状、分類できていない節は存在しないが、定義節は細分化されるべきである。
- 定義節とは性質が異なる「制約節」として新たに定義されるべきである。
- 処理節と制約節は意識して明確に区別されている必要がある。
- 状態遷移モデル中に現れないことから、本来条件節とも言える箇所を制約節は包含して持つ。

3.3 全体の考察

RQ1の結論から、解析ツールは今回対象とした要求仕様書については、非常に高い精度で節の分類を行うことができる。また、RQ2の結論から、今回対象とした要求仕様書については、制約節を種別として追加すると、すべての文を節に分類することが可能となった。

3.2においても述べたが、制約節として記述されているか、処理節として記述されているかは、書き手の意思を反映している可能性がある。そのため、制約として

既述されているだけなのか、それとも処理としてイベントに応じて実行されるべきなのかは、節の種別を明確に表す書かれ方によって決定される。このことから、解析ツールによる対応の可否だけではなく、より高い品質の要求仕様書を作成する上でも、制約か処理かを明確にすることは重要であると言える。

精度の高い節の分類は、解析ツールが最終的に目的としている状態遷移モデルの抽出だけではなく、様々な応用が考えられる。たとえば、形式手法の記述支援に繋がる可能性がある。処理節と条件節は、そのまま形式モデルにおける対応する表現として記述でき、具体的な定義や制約が型宣言や不変条件、事前条件、事後条件、時相論理式による検査式など、要求仕様書を形式モデルに書き換える際に注視する箇所を明らかにできる可能性がある。大森らによって、自然言語からキーワードを抽出することで形式モデルの作成支援を行うツールが開発されている [7]。

また、ソフトウェアテストにおける、テストケース作成支援にもつながる。条件が明らかになるため、入力パターンを作成することが容易となり、それに対する出力（振る舞い）も対応する処理節を確認すればよい。また、非機能要求が制約節として書かれていれば、機能要求以外の要求を抽出することも容易になる可能性がある。

そのため、解析ツールにおける節の分類の評価を、以下のようにまとめる。

- 内容を確認した結果、高い精度で節を分類することができる。
- 新たに制約節を定義することで、節の分類としては現状十分である。
- 状態遷移モデル以外の応用の可能性がある。

3.4 妥当性への脅威

本稿における妥当性への脅威を説明する。まず、実験で対象とした要求仕様書が、解析ツールの制約を満たすように書き換えられたものである。加えて、解析ツールの開発者である第2著者が書き換え作業の確認を実施していたことがあり、結果が良いものとなったことに影響を与えている可能性がある。また、実験で対象とした要求仕様書は、箇条書きを参照する際、すべて「以下の」という表現を持っていたが、箇条書きを参照する場合には「次の」や「下に示す」など、他の表現も存在する。このように、ツールで解析可能となるよう対象の文書を修正しているため、もし、想定されていない記述が存在した場合には精度が低下する可能性がある。

他に、正解集合の作成およびインタラクティブな操作を第1著者が実施していることで、より高いF値が得られた懸念がある。

また、実験で用いた要求仕様書は1つだけであるため、異なる要求仕様書を用いると結果が異なる可能性がある。

4 関連研究

自然言語によって記述された文書から様々な情報を抽出する研究が、多数存在する。村上らは、自然言語で記述されたシステム仕様書から、試験ケースを作成するまでのプロセスを提案している [8]。彼らは、セミ形式記述を定義し、文章をセミ形式記述に変換するアルゴリズムを定義し、条件・動作同定機能を実装したツールを開発した。条件、動作は、いずれも8割以上の *precision* および *recall* によって抽出されている。彼らは、テスト支援を目的としており、我々は状態モデル作成支援を目的としている。他に、要求仕様書からドメイン固有言語 (DSL) を抽出支援する手法がある [9]。彼らは、自然言語とそれに対応する DSL、および DSL の文法を機械学習し、自然言語から DSL を生成するフレームワークを提案している。また、自然言語のユーザマニュアルからソフトウェアの機能に関連する情報を抽出支援する手法が提案されている [10]。

自然言語で記述された文書の誤りを抽出する研究も存在する。辞書を用いて書き誤る可能性が高い日本語助詞の検出・修正を行う研究が存在する [11] [12]。今後、これらの技術を用いて、入力する文章の制約を緩和可能か検討する。これらの技術を

利用することで、我々の研究におけるツールが対応すべき制約を緩和できる可能性がある。

また、第1著者の山本をはじめ、C言語ソースコードから状態遷移表を抽出する支援ツールの開発を進めてきた [13]。本稿における解析ツールによって生成された状態遷移モデルと、ソースコードから状態遷移表を抽出するツールの生成結果を比較することで、要求仕様書とソースコードの整合性を確認できる可能性がある。

5 おわりに

先行研究において提案された手法を実装したツールに対して節の分類精度の調査を行い、今回対象とした要求仕様書では高い精度で節を抽出・分類できることがわかった。しかし、今後、定義節を細分化して制約節を検討する必要があるとわかった。節の分類は、状態遷移モデル抽出だけではなく、その他のモデルや、テスト支援、要求の整理などにも活用できる可能性があり、様々な応用が可能であると考えている。今後の課題は、実験参加者を増加させるなどのより妥当性を向上した実験の実施、他の要求仕様書を与えたときの性能を調査すること、および分類精度の向上をすることである。また、現状、F値のみに関する評価のみ実施しているが、その他の評価方法を検討して解析ツールを評価する必要がある。本稿のツールの評価について、節の分類が状態遷移モデル生成に与える影響の評価も今後必要である。

参考文献

- [1] 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.
- [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] Ekaterina Boutkova and Frank Houdek. Semi-automatic identification of features in requirement specifications. In *Proc. of RE 2011*, pp. 313-318, 2011.
- [4] 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-SE-198, No. 5, pp. 1-8, 2018.
- [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. 名古屋大学大学院情報学研究科 情報システム学専攻 修士論文, <https://sites.google.com/site/yoshidaatnu/naru.pdf>, 2019.
- [6] 中村成, 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を対象とした状態遷移モデル作成支援. 電子情報通信学会技術研究報告, Vol. 118, No. 230, pp. 25-30, 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, Vol. 3, No. 5, pp. 18-28, 2010.
- [8] 村上響一, 青山裕介, 村上神龍, 久代紀之, 牧茂, 田畑一政, 神代勉, 中村潤. 自然言語仕様書からの試験ケース生成のための条件・動作の同定手法. 情報処理学会研究報告, Vol. 2018-SE-198, No. 7, pp. 1-7, 2018.
- [9] Aditya Desai, Sumit Gulwani, Vineet Hingorani, Nidhi Jain, Amey Karkare, Mark Marron, and Subhajit Roy. Program synthesis using natural language. *Proceedings of the 38th International Conference on Software Engineering, ICSE '16*, pp. 345-356, 2018.
- [10] Thomas Quirchmayr, Barbara Paech, Roland Kohl, Hannes Karey, and Gunar Kasdepke. Semi-automatic rule-based domain terminology and software feature-relevant information extraction from natural language user manuals. *Empirical Software Engineering*, Vol. 23, No. 6, pp. 3630-3683, 2018.
- [11] 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, Vol. 2007-NL-181, No. 94, pp. 107-112, 2007.
- [12] 今枝恒治, 河合敦夫, 石川裕司, 永田亮, 榊井文人. 日本語学習者の作文における格助詞の誤り検出と訂正. 情報処理学会研究報告, Vol. 2003-CE-068, No. 13, pp. 39-46, 2003.
- [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抽出支援ツール. 電子情報通信学会論文誌 D, Vol. J102-D, No. 3, pp. 151-162, 2019.