Vol.2013-SE-181 No.17
2013/7/18

自然言語解析を用いた要求仕様の評価手法

弓倉陽介[†1]　和田大輝[†1]　鷲見毅[†1]　藤本宏[†1]　村田由香里[†1]

近年，要求仕様の品質を向上させるために，自然言語解析を用いて文章校正や分析用のモデル自動生成が試みられている．しかし，これらは記述されている要求仕様の曖昧性や誤りを排除することには効果的だが，暗黙的な要求仕様を前提とした記述を発見するために利用することは難しい．そこで本研究では，自然言語解析結果から得られる文の構造的情報以外の付加的情報であるモダリティ情報を利用し，要求仕様から抜け漏れた可能性のある情報をダイアグラムの形式で提示する手法を提案する．

Applying Natural Language Analysis to the Evaluation of Requirements Specifications

YOSUKE YUMIKURA[†1]　TAIKI WADA[†1]　TAKESHI SUMI[†1]
HIROSHI FUJIMOTO[†1]　YUKARI MURATA[†1]

Nowadays natural language analysis techniques are employed in revision support and automatic generation of analytical models with the aim of quality improvement of requirements specifications. These techniques are effective in eliminating ambiguities and errors in requirements specifications but are not of much help in detecting deficiencies therein. In this paper we will present a method of detecting deficiencies in requirements specifications and of displaying them in diagram form, which makes use of not only sentence structures but also sentence modalities obtained by natural language analysis.

1. はじめに

近年，システム開発では要求仕様の品質向上がますます重要になってきている．特に，大規模なシステムの構築では，要求仕様の不備が手戻りコストの増大につながる．大規模化したシステムでは多様なステークホルダが関係することが多いため，一意に理解できる要求仕様が必要となってくるため，自然言語にて記述された要求仕様の品質向上が望まれる．

要求仕様の品質はレビューにて評価しているのが一般的であるが，要求仕様の品質をレビューだけで向上することは難しい．一つは，レビュアの要求仕様に対する理解を容易にするために，要求仕様の記述者が各種ダイアグラムや表を用いる場合，その作成コストが大きくなることが問題となる．また，レビュア毎に能力が異なり，要求仕様の品質が向上するかどうかは属人的になることも問題となる．そのため，自然言語で記述された多量の要求仕様の品質評価が機械的に行える自然言語解析技術に注目が集まっている．

従来の自然言語解析技術を用いた要求仕様の品質向上技術では，要求仕様の曖昧性や誤りについて指摘することに重点が置かれてきた[2]．例えば，自然言語解析により要求仕様内に登場する用語や文の修飾関係の曖昧性を検出することが研究されている[3-5]．また，自然言語解析した結果から UML の各種ダイアグラムやイベント発生の流れ図の生成を試みるものも存在する[6, 7]．しかし，これらの研究では要求仕様の抜け漏れの問題，すなわち，暗黙的な要求仕様を前提とした記述が引き起こすシステムの不具合を解決することが難しい．これらの研究では文の構造や用語の意味を利用している一方で，要求仕様の文が暗に意味していること，つまり，記述の意図を担う要素を汲み取れていないためである．

これまで我々は，要求仕様の抜け漏れを検出するために，自然言語から DFD（Data Flow Diagram）を生成する手法を提案してきた[1]．要求仕様の文から生成された DFD は，要求仕様の抜け漏れによって DFD の構文規則に違反したものが生成されることや，特徴的な構造になるため，視覚的に要求仕様の抜け漏れを検出することができる．しかし，これまでの DFD 生成方法では記述の意図を担う要素が汲み取れていなかったため，生成される DFD はユーザが意図しないものになることがあった．本研究では，記述の意図を担う文中の強調表現や機能語や付属語の用い方などのモダリティと呼ばれる情報を取り入れることで，要求仕様の意図を正しく反映した DFD を生成し，抜け漏れの検出をより容易にできるようにする手法を提案する．

2. 要求仕様の抜け漏れ検出の課題

要求仕様の抜け漏れには，２つの種類が存在する．一つが暗黙の要求仕様と呼ばれるもので，要求仕様の記述者が前提知識として既に合意しているものと仮定して記述しな

†1 (株)東芝ソフトウェア技術センター
Toshiba Corporation, Corporate Software Engineering Center

かったものを意味する.もう一つが考慮不足だった仕様で,要求仕様の記述者のミスによる記述漏れを意味する.

要求仕様の抜け漏れについての具体例をポットの要求仕様を用いて説明する．なお，本稿では要求仕様の例として『話題沸騰ポット GOMA -1015 型要求仕様書』第 7 版を利用，および，参考にする[8].

暗黙の要求仕様の例をポットの要求仕様の例で示すと，「高温モード，節約モード，ミルクモードが保温モードとなる．節約モードのときは，保温時の目標温度を９０℃とする.」というような要求仕様である．この要求仕様では，節約モード以外での目標温度の記述がされておらず，製品知識やドメイン知識の少ない開発者がこの要求仕様から開発を行うことは不可能である.

考慮不足の要求仕様の例は，加熱処理に関する要求仕様が以下しか記述されていないような状況である.

「加熱時は，目標温度の＋１℃まで加熱する.」

この要求仕様では加熱後の処理について全く記載されていない．つまり，ポットであるにもかかわらず保温する処理が要求仕様から抜け落ちてしまっている．このような要求仕様が考慮不足の要求仕様となる.

これらの問題は，要求仕様の量が増大することで更に発生する可能性が高くなる．要求仕様の記述者が作業量の増加に伴い，当然の要求仕様とみなして暗黙の要求仕様を発生させてしまうことや，量の多さにより要求仕様の項目間の関係性が十分に検討できず考慮不足の要求仕様を発生させることを助長してしまうためである.

3. 自然言語解析を用いた DFD 生成手法の課題

我々は，2 章で示したような要求仕様の抜け漏れを検出するために，自然言語の格関係と係り受け構造を用いて DFD を生成する手法を提案しており[1], この手法により生成される DFD の以下の様な特性を検討することにより, 要求仕様の抜け漏れの検出を試みている.

- **無名の要素（図 1）:** DFD 要素が存在する可能性がある場合，その要素が存在しうる場所に無名の DFD 要素が生成される．この無名の要素が何であるかを検討することで, 要求仕様の抜け漏れに気づくことができる.

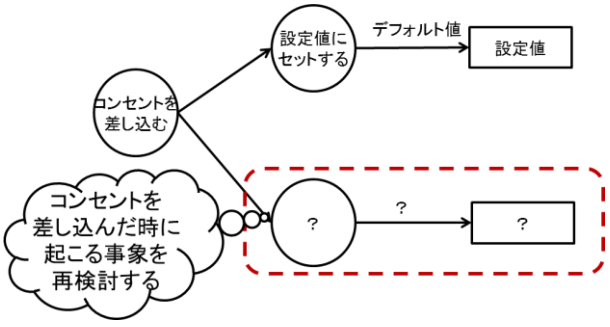


図 1：無名の要素による抜け漏れ検討

- **DFD の断裂（図 2）:** 要求仕様から DFD を作成するときに，DFD の階層が同じ場合，作成される DFD は断裂することはないはずである．つまり，エッジの向きを考慮しなければ任意のノードからそのノード以外の全てのノードへたどり着くことができると考えられる．そのため，生成される DFD が断裂した場合, 要求仕様が抜け漏れている可能性があり，その点について考察することで，要求仕様の抜け漏れに気づくことができる.

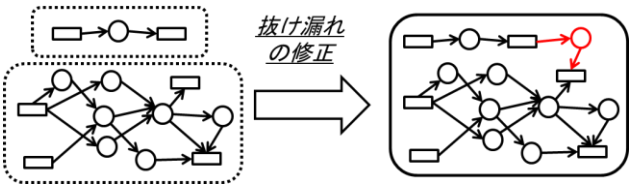


図 2：DFD の断裂による抜け漏れ検討

しかし，従来の手法では要求仕様の文が意図している内容を正しく汲み取った DFD を生成することができなかった．例えば，要求仕様の抜け漏れを検出する上で本質的でない無名の要素しか生成されないことや，DFD のグラフ構造が人の予想と異なった形状で生成される場合があり，本来気づくべき抜け漏れとは無関係な抜け漏れを検出してしまうという問題があった.

4. モダリティを用いた DFD 生成手法

まず，本手法の概要を説明する．その後，いくつかの点について詳細を説明する.

4.1 概要

本手法では, 自然言語から DFD を生成する際にモダリティを利用することで, 従来の DFD 生成手法の課題を解決する．モダリティとは，文が表す客観的な事柄内容以外の情報であり，主に記述者の心的態度や事柄間の時間的な関係といったもの指す[13]．これまで，言語処理学の領域で発話者の心的態度を解析するための重要な情報として利用されてきたが[14]，要求仕様の文章の評価にはまだ応用されていない．モダリティが文に付与されることで，文の示す事柄内容に記述者の心的態度や事象の時間的関係などの情報を付与できる.具体例として,「水温が９０℃であり, ～」と「水温が９０℃である場合，～」という２つの文について比較する．前者はモダリティを特に持たない文であり，「～」において水温以外の値について述べようとしている．一方で後者では「～場合, 」がモダリティを含んでおり,「水温が９０℃である」ことを条件として「～」が発生することを意図している．このように同一の構造を持つ文においても，そこに付与されるモダリティによって意図する内容が変わってくるため，モダリティを考慮することで，人の記述した意図を上手く反映したモデルを生成することができる.

本手法では次の 4 つのステップの中でモダリティを利用した DFD 生成手法について述べる（図 3）.

Step 1: 自然言語解析による要求仕様情報の抽出

形態素解析，係り受け解析，モダリティ解析を行い，これらの解析結果から DFD モデルの生成に利用する，出現語，文構造，モダリティの情報を抽出する.

Step 2: 要求仕様情報から DFD 構造の生成

Step 1 で取り出した情報から DFD 構造を生成する．具体的には，出現語情報と文構造情報から決定されるルールに応じて DFD 構造を生成する.

Step 3: モダリティを利用した DFD 構造の変形

Step 2 で生成された DFD 構造を保有するモダリティ情報に従って変形する．具体的には，モダリティ情報から用いる DFD 構造の変形ルールが決定される.

Step 4: DFD 構造の統合

要素名と単文構造間の係り受け関係を用いて，作成されている DFD 構造を統合し，ひとつの DFD を構築する.

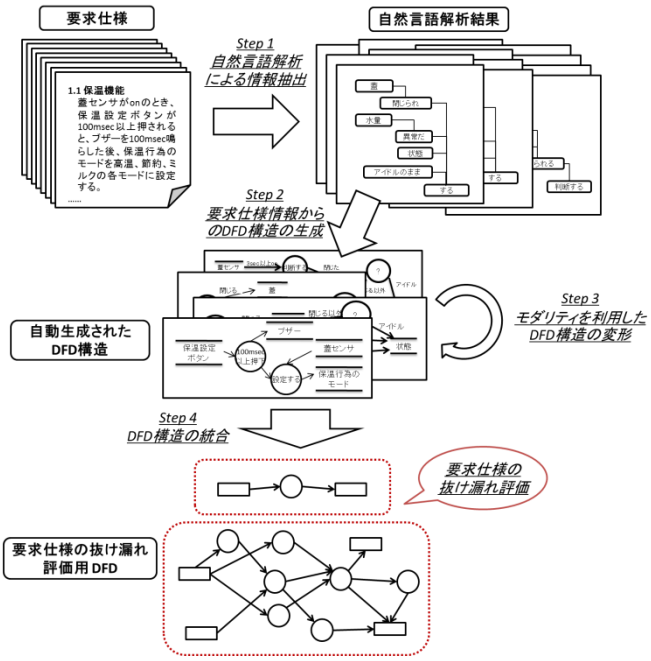


図 3：要求仕様の抜け漏れ評価手法の外観

次節からの説明では，文献[8]に含まれる以下の要求仕様 RS1 を利用する.

【RS1】*蓋が閉じられても、水量が異常な場合、状態はアイドルのままである。*

4.2 Step 1: 自然言語解析による要求仕様情報の抽出

DFD 構造を生成するために必要な情報について述べたあと，本手法の特徴である，モダリティ解析について述べる．また，最後に DFD 構造へのマッピングに利用する単文構造への分解についても述べる.

4.2.1 DFD 構造生成に必要な情報

DFD の生成に必要な情報を抽出する．ここでは，以下の 3 つの情報を抽出する（図 4）.

- 出現語情報：文章内に登場した語，及び，その語の属性情報．この情報を取得するために，形態素解析[9]を用いる.
- 文構造情報：出現語が形成する文節間の係り受け関係情報．この情報を取得するために，係り受け解析[9]を利用する.
- モダリティ情報：文節が持つ文構造情報以外の情報であり，文節，及び，文節の組み合わせの意味を変化させたり，追加したりするもの．例えば，ヴォイス，ムード，アスペクト，テンス，などがこの情報にあたる[10]．この情報を取得するために，文に含まれるモダリティ情報を抽出する．これをモダリティ解析と呼ぶ.

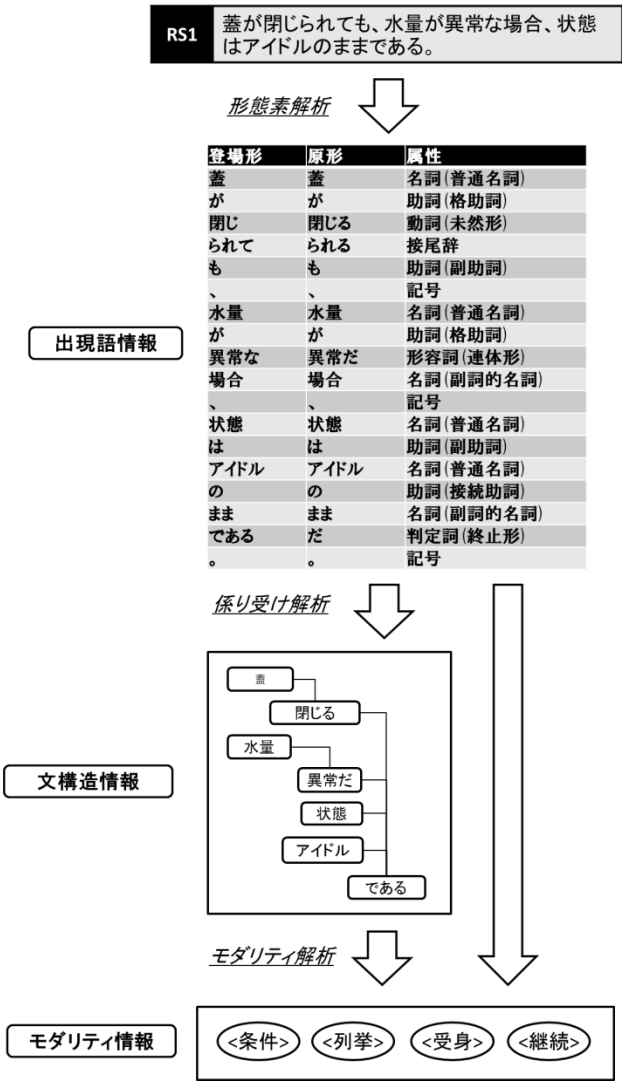


図 4：要求仕様情報の抽出と抽出物

4.2.2 モダリティ解析

モダリティ解析とは，出現語情報と文構造情報を用いて文節，または，文節の組み合せが保持しているモダリティ

情報を判定する解析のことである．これにより，要求仕様から構造情報以外の情報を抽出する．例えば，「蓋が閉じられても、」という文からは「動詞（未然形）＋接尾辞：られる」という出現語情報と「閉じる」が「である」に係るという文構造情報が抽出できる．この情報から，この文を構成する文節の組み合せにモダリティ＜受身＞が存在することがわかる．これ以外に解析できるモダリティの例を表 1 で提示する．

表 1：モダリティ種別とその解析ルールの例

モダリティ種別	モダリティ解析ルール
＜受動＞	「動詞(未然形)」＋「接尾辞:られる」
＜提題＞	「副助詞:は」
＜列挙＞	「副助詞:も」
＜条件＞	「名詞:場合」
	「動詞(仮定形)」
＜限定＞	「係助詞:のみ」
	「係助詞:しか」
＜過去相化＞	「時相名詞:以前」＋「接続助詞:の」
＜禁止＞	「形容詞:いけない」
＜完了＞	「動詞(連用テ形)」＋「接尾辞:しまう」
＜継続＞	「副詞的名詞:まま」
	「動詞(連用形)」＋「動詞:続ける」
	「動詞(連用テ形)」＋「接尾辞:いる」

4.2.3 単文構造への分解

図 5 に示すように，要求仕様の情報から DFD を生成するために，要求仕様の情報を単文構造の粒度で切り分けることを行う．単文構造とは，文に存在する述語が 1 つだけの文を指し，述語に他の述語が係らない構造である．

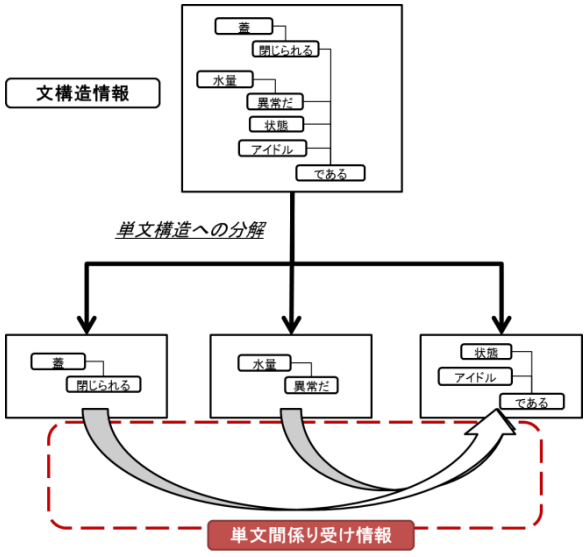


図 5：単文構造への分解

このとき，複文となっている要求仕様は単文に分解されるが，複文のときに存在した単文間の係り受け関係を後の処理で利用する必要がある．このため，この単文間の係り受け関係を単文間係り受け情報として取り扱う．これにより，DFD 生成後に述語の係り受け関係から DFD 要素間の接続関係を構築する．

4.3 Step 2: 要求仕様から DFD 構造の生成

Step1 で抽出した単文構造の情報から DFD 構造へのマッピングを行うことで単文構造毎の DFD 構造を生成する．その手順について以下で詳説する．

4.2.3 項で生成された単文構造に対して DFD 生成ルールを適用し，単文構造の情報から DFD を生成する．DFD 生成ルールは，単文構造の持つ述語と格関係を DFD の構成要素とその結合方法にマッピングする形で記述されている．

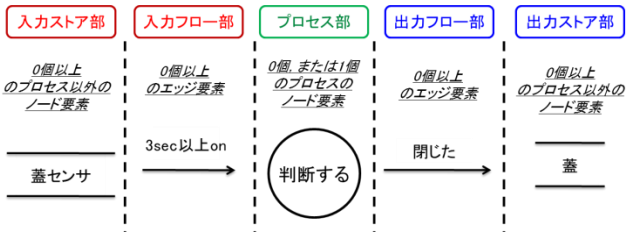


図 6：DFD 構造

どのような単文構造も必ず以下の構成部の組み合せとして表現される（図 6）．

- **プロセス部：**0 個以上のプロセス
- **入力ストア部：**0 個以上のデータストア，または，ターミネータ
- **入力フロー部：**0 個以上のデータフロー．ソースはプロセス以外の要素となり，ターゲットはプロセスとなる．
- **出力ストア部：**0 個以上のデータストア，または，ターミネータ
- **出力フロー部：**0 個以上のデータフロー．ソースはプロセスとなり，ターゲットはプロセス以外の要素となる．

このようなマッピングが可能となるのは，DFD の構成要素であるプロセスは述語で記述されるものであり，単文構造には述語が必ず一つしか存在しないためである．

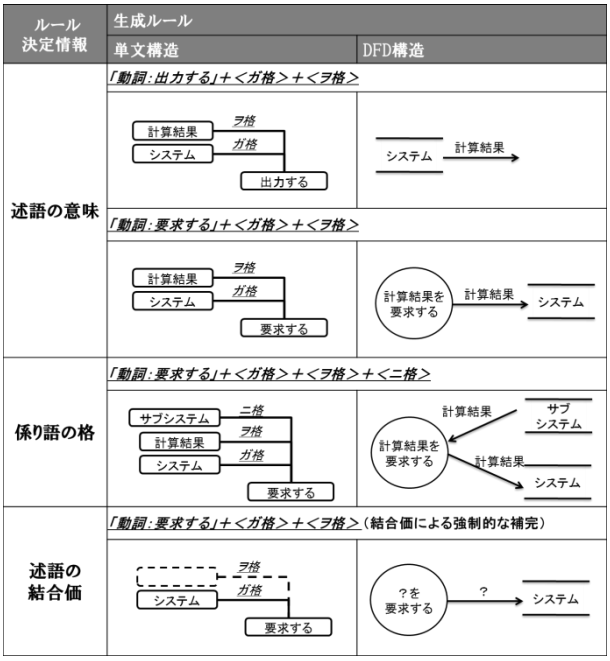
実行される DFD 生成ルールは単文構造の以下の情報に基づいて決定される（表 2）．

- **述語の意味：**生成される DFD の構造は，述語の意味によって決定する．例えば，単文構造が「A が B を～する」という構造であっても，「システムが計算結果を出力する」と「システムが計算結果を要求する」では DFD の構造は異なることが望ましいためである．
- **係り語の格：**文に登場する語が文の意味の中で果たす役割は述語に係る際の格によって決定される．この係る際の格に応じて，生成される DFD の構成部への割り当てが決定する．例えば，「システムが計算結果をサブシステムに要求する」という単文構造に対して，ガ格の「システム」は出力ストア部に，

ヲ格の「計算結果」は出力フロー部と入力フロー部に，二格の「サブシステム」は入力ストア部に割り当てられる．

- **述語の結合価：**文に登場する述語の意味を考慮したときに，必ず省略できない格関係が存在する．そのような関係を結合価と呼び，単文構造においてその情報が欠落している場合には無名の要素として，欠落している格の要素を生成する．例えば，「システムが要求する」という文章では，要求しているものが抜け落ちている．このような場合，DFD を生成する際に，無名のデータフローで「システム」という要素名のデータストアと「要求する」というプロセスを接続する．

表 2：DFD 構造生成ルール例



4.4 Step3: モダリティを利用した DFD 構造の変形

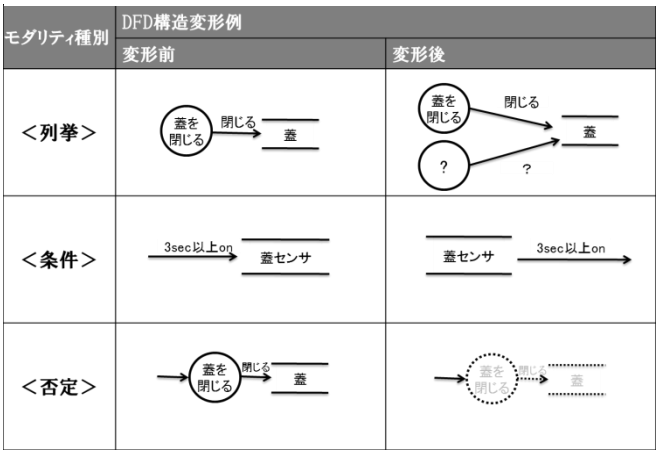
要求仕様の意図することを表現した DFD にするため，単文構造毎に保持していたモダリティ情報を用いて 4.3 節で生成された DFD 構造の変形を実施する．

DFD 構造を変形するために，図 6 の構成部，または，構成部と出元となった格を指定して変形ルールを記述する．

その変形ルールの例を示す．

- 列挙モダリティルール：このモダリティは他の同じような働きの要素が別に存在することを示唆する．そのため列挙モダリティが属する文節と同じ構成部に名前が「？」の無名の要素を作成する．表 3 の１行目は「蓋を閉じても，」という，列挙のモダリティを持つ文から DFD を生成・変換した結果である．「～ても」という表現が列挙のモダリティを文に与えている．
- 条件モダリティルール：このモダリティは単文構造では入力フロー部になる要素を出力フロー部に，入力ストア部を出力ストア部に変更する．表 3－2 行目は「蓋センサが 3sec 以上 on のとき，」という条件モダリティを持つ文から DFD を生成・変換した結果である．「～のとき」という表現が条件モダリティを文に与えている．
- 否定モダリティルール：このモダリティはプロセス部が実施されないことを意味するため，プロセス部，および，その出力側の要素を削除する．表 3 の 3 行目は「蓋を閉じないで，」という否定モダリティを持つ文から DFD を生成・変換した結果である．「～ないで」という表現が否定モダリティを文に与えている．

表 3：モダリティによる DFD 構造の変形例



4.5 Step 4: DFD 構造の統合

全ての DFD 構造を統合し，ひとつの DFD にする．このとき，DFD 構造間の接続には以下の 2 つを利用する．

- **単文間係り受け情報：**Step 1 の単文構造への分割時に作られたデータ．DFD 構造間にこのデータが存在する場合，2 つの DFD 構造をデータフローによって接続する．
- **要素名：**複数の DFD 構造において要素名と要素種別（プロセス，データストア，ターミネータ）が同じ場合，同一の要素とみなし統合する．

この処理によって生成される DFD を用いて，3 章の基準に従って，要求仕様の抜け漏れを人手で評価する．

5. 適用事例

本手法の適用事例としてポットの要求仕様[8]の一部から生成した DFD を提示する（図 7）．これを用いて，今回の生成手法により生成される DFD と従来手法で生成された DFD との違いについて説明する．変換する要求仕様として要求 ID が pot-220 のものを利用する（表 4）．

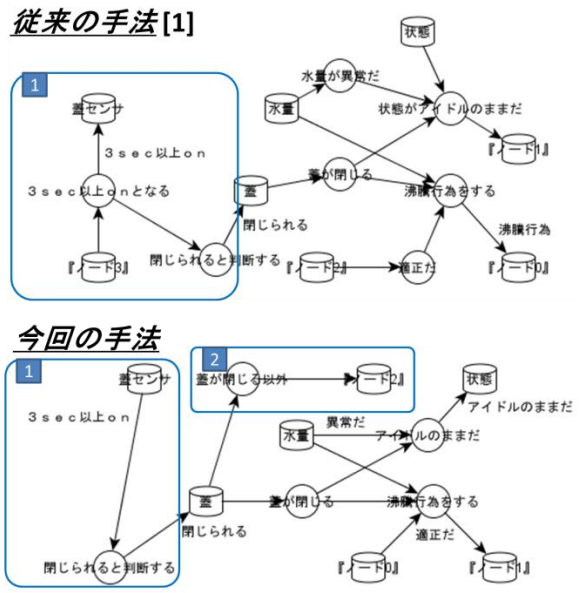


図 7：従来手法で生成した DFD と本手法で生成した DFD

まず，図 7 内の枠番号 1 については同じ要求仕様（pot-220-11）から生成されたものである．この要求仕様の文には，「～なったら」という語により条件モダリティが付与される．従来手法では条件モダリティが無視されるため，「3sec 以上 on となる」プロセスになんらかの入力があると仮定して，無名の「『ノード 3』」が生成される．一方本手法では，条件モダリティにより「3sec 以上 on」がデータフローとして表され，不要な無名ノードが生成されない．

表 4：DFD の生成元となる要求仕様[8]

要求	pot-220	アイドルの状態で、蓋を閉じたら、水位を確認し、条件に合えば沸騰行為をする。
	理由 **説明**	蓋を閉めるという行為で加熱(沸騰)の指示をしたい。 沸騰行為の詳細は、3 章の「温度制御行為」に記載する。 蓋センサが on になって 3sec 経過するのを末理由は、注水やポットの移動の直後に、水面が波打っている状況が考えられるので、水面状態が安定する時間を想定したためである。
□	pot-220-11	蓋センサが 3sec 以上 on となったら、蓋が閉じられたと判断する。
□	pot-220-21	蓋が閉じられ、水量が適正な場合、沸騰行為をする。
□	pot-220-31	蓋が閉じられても、水量が異常な場合、状態はアイドルのままである。

次に，図 7 内の枠番号 2 については本手法でのみ生成されているものである．この要素は要求仕様 pot-220-31 から生成されてきている．この要求仕様の文には，「～も」という語により列挙モダリティが付与されている．そのため，生成ルールによって「蓋が閉じられている」以外の要素として存在する可能性のある「蓋が閉じる以外」というプロセスを生成する．このように，本手法では要求仕様の抜け漏れが存在する可能性の高い要素を提示することで，DFD の閲覧者に要求仕様が抜け漏れていないか気づきを促せる．

評価と考察

本手法を評価するために，文献[8]の 2 章と 3 章の全ての要求仕様から DFD を生成する．ただし，この要求仕様には様々な文献において、仕様の不備が指摘されており，中でも用語の揺らぎに対する指摘が多い[11]．本手法では 3.5 節に記載した Step 4 の処理のために，用語の揺らぎにより，生成する DFD が断裂する．今回の評価では要求仕様の抜け漏れをチェックすることが目的であるため，用語の揺らぎを事前に取り除くことが望ましい．そのため，以下の DFD 生成においては，仕様不備の指摘を修正した要求仕様[11]に対して．以下の 2 点について評価と考察を行う．

1. 本手法で生成した DFD を要求仕様の意図をどの程度適切に反映しているかの観点で評価する．
2. 本手法で生成した DFD から要求仕様の抜け漏れを検出する．

5.1 生成 DFD の比較

従来手法と本手法のそれぞれで生成した DFD を比較する．それぞれの手法で生成された DFD の各種計測値は以下の表 5 のとおりである．

表 5：生成された DFD に関する計測値

計測数	従来手法	本手法
ノード数	232	185
エッジ数	295	262
無名のノード数	68	47
クラスタ数（断裂）	4	5

無名のノード数の減少は予測できていたとおりであった．この理由は，条件モダリティルールなどの，DFD の接続を文の意図に合わせて変形させることで，無駄な無名のノードの生成を抑制できたためと考えられる．

生成された DFD のノード数は無名のノード数が減少するよりも多い量の減少が見られる．この理由は，無駄な無名のノードの減少理由と同じだけでなく，否定モダリティルールのような DFD の要素を削除するルールにより，無名ではないが，要求仕様を DFD として表現する上で不要な DFD 要素を削除することができたためと考えられる．それに対して，エッジ数の減少量が少ないのは，上記のように要素の統合によるノード数の減少に関係する．ノードの統合によりノード数は減少するが，統合前のノードに結合していたエッジはそのまま統合後のノードに結合しつづける．

クラスタとは，DFD が断裂せずに結合している要素の集合のことであり，クラスタ数とはこの DFD の断裂によってできたクラスタの数を意味する．本手法で生成した DFD よりも従来手法で生成した DFD の方がクラスタ数は少ない．この理由は，従来の DFD 生成手法では，表 5 のエッジの数が多いことからもわかるように，意図しないエッジが作成され，結果 DFD が断裂せずに意図しない結合をしているものがある．本手法で増えたクラスタが要求仕様の抜け漏れに関与するかどうかについては 6.2 節で再度検討する．

これらより，本手法が従来手法よりも DFD 生成時に不要な要素の生成を抑制できており，要求仕様の抜け漏れを検討する際，適切な DFD を用いることによって，余分な分析工数を減少させることができると考えられる．一方で，生成される DFD の規模が分析に利用するには大き過ぎるという課題がある．本手法で生成される DFD のノード数が DFD を分析する上で理解しやすいノード数として挙げられている 7±2 という値から大きく離れている[12]．この問題に対しては，フィルタリングやクラスタリングを用いて仕様の可視化方法を工夫する必要がある．

5.2 要求仕様の抜け漏れ検討

本手法によって生成された DFD を用いて要求仕様の抜け漏れを分析した結果，要求仕様[11]に対して，以下の抜け漏れが存在する可能性を示唆できた．

- **無名の要素からの検討:**「給湯ボタンを離した時のポンプの状態が不明」という要求仕様の抜け漏れを検出できた．変換に関係した仕様は以下の２つである．
  ・「以下の条件を全て満たすとき，給湯ボタンを押している間，ポンプを給湯中とする．」
  ・「ポンプが給湯中のときに，一つでも以下の条件になったとき，給湯ボタンを押している間でも，ポンプを停止する．」
  今回は特殊な無名のノード「給湯ボタンを押す以外」というプロセスが「～でも」という語が持つ列挙モダリティにより生成される（図 8）．ここから，ユーザが給湯ボタンを押した時に，ポンプが給湯中であることは要求仕様から読み取れるが，ユーザが給湯ボタンを離したときのポンプの状態が要求仕様において未定義であることがわかった．

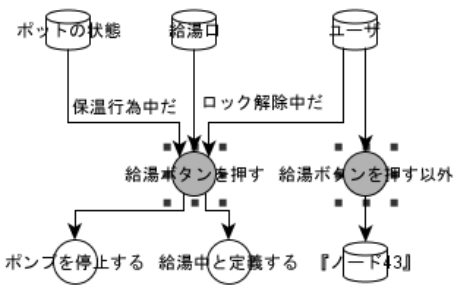


図 8 :「給湯ボタンを押す」の隣接ノード

- **グラフの断裂に対する検討:**「コンセントをつないだときにタイマ残り時間表示窓にタイマ残り時間・分を表示する／しない」という要求仕様の抜け漏れを検出できた．生成された DFD をグラフが結合しているかどうかでクラスタリングした結果とその時に生成されたクラスタ群が以下の図 9 のとおりとなる．

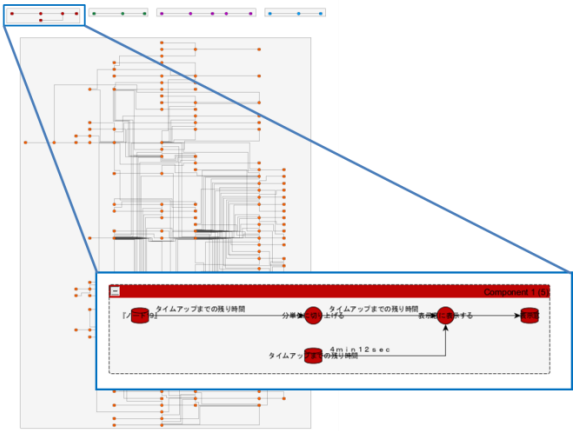


図 9 : 断裂を起こした DFD 要素のクラスタ

生成されたクラスタで断裂している小規模なものに着目すると，「タイマ残り時間表示窓」という要素が存在する．この要素を含むクラスタは，他の要求仕様から生成された DFD 要素が形成するクラスタと異なっており，コンセントの要素とタイマ残り時間表示窓の要素が断裂している．このことから，コンセントが繋がれた時にタイマ残り時間表示窓に残り時間 0 分を表示するかどうかの要求仕様が抜け漏れていることを本手法によって示唆できた．

このように，文献[11]の要求仕様に対して本手法を用いて要求仕様の抜け漏れがある可能性を示唆できた．これは，モダリティの利用により，無名の DFD 要素が誤生成されることを抑制でき，抜け漏れを検討すべき対照の捕捉が容易になったことと，要求仕様の記述者の意図を汲み取り，抜け漏れている可能性が高いことを示す DFD 要素を提示できたことによる．このため，従来手法よりも本手法が要求仕様の抜け漏れの検出を容易にする事の効果が示せた．

6. 関連研究

自然言語解析を利用した要求仕様の品質向上をテーマとした関連研究には以下の様なものが存在する．

まず，自然言語解析の結果を用いて要求仕様の文章に直接指摘を与える研究がある[3-5]．文章の係り受け構造の曖昧性を指摘するもの[3]，代名詞の曖昧性を指摘するものがある[4]．これらの研究はいずれも記述されている内容に対して指摘する研究であり，本研究とは解決する要求仕様の問題が異なる．また，特定の構文パターンにマッチするか

を調べることで，主語や目的語が抜けていることを指摘する研究が存在するが[5]，これは記述が曖昧になっていることを指摘するものであり，本研究のように要求仕様の抜け漏れを示唆するものではない．

自然言語解析の結果を形式的な構造に落とす研究も存在する[6, 7]．構文解析した結果を人手で UML モデルにマッピングするもの[6]，揺らいでいる用語を統合した後に流れ図にマッピングするものが存在する[7]．本研究は，自然言語解析した結果を DFD というモデル要素にマッピングしているが，その生成時にモダリティの情報を用いていることで，要求仕様の記述の意図をより詳細に拾い上げている点で異なる．

7. おわりに

要求仕様を自然言語解析した結果から DFD を生成すると，従来手法においては，ユーザが意図しない DFD 構造が生成される場合があった．本研究では，要求仕様の文が含むモダリティの情報も活用するようにした新たな DFD 生成手法を提案した．これにより，要求仕様の抜け漏れを従来手法より容易に検出できる DFD を生成することが可能である．

今後の課題としては，生成される DFD が大規模になり要求分析に利用し難い場合がある．これに対して，DFD の自動階層化や要求仕様の抜け漏れ検出の自動化手法の開発により対応していく予定である．

参考文献

1) 弓倉陽介，鷲見毅，藤本宏，和田大輝，村田由香里: 段階的形式化手法による要求仕様の品質向上,ウィンターワークショップ 2013・イン・那須 (2013).
2) 竹内広宣，中村大賀，萩野紫穂，水野謙，岩間太，鎌田真由美:ソフトウェア開発における文書成果物の分析技術とその活用，コンピュータソフトウェア, Vol.30, No.1, (2013), pp.53-64.
3) Y. Hui, W. Alistair, R. Anne, and N. Bashar: Automatic Detection of Nocuous Coordination Ambiguities in Natural Language Requirements, Proc. IEEE/ACM International Conference on Automated Software Engineering, pp.53-62 (2010).
4) Y. Hui, R. Anne, G. Vincenzo, W. Alistair, and N. Bashar: Extending Nocuous Ambiguity Analysis for Anaphora in Natural Language Requirements, *Proc. 18th IEEE International Requirements Engineering Conference*, pp. 25-34 (2010).
5) 有賀顕，林晋平，佐伯元司: 構文と文章構造に基づく要求仕様書の問題点発見支援，情報処理学会研究報告, vol.2013-SE-179, no. 4 (2013).
6) R. Drechsler, M. Soeken, and R. Wille: Formal Specification Level: Towards Verification-Driven Design Based on Natural Language Processing, *Proc. Forum on Specification and Design Language*, pp. 53-58 (2012).
7) 菊川正人，白銀純子，深澤良彰: 自然言語解析技術を応用したイベントフローの統合によるプログラム理解，信学技報, vol. 102, no. 602, pp. 31-36 (2003).
8) 話題沸騰ポット（GOMA-1015 型）要求仕様書第 7 版，組込みソフトウェア管理者・技術者育成研究会 (2005).
9) 天野真家，石崎俊，宇津呂武仁，成田真澄，福本淳一:自然言語処理，オーム社 (2007).
10) 益岡隆志，田窪行則:基礎日本語文法―改訂版―，くろしお出版 (1992).
11) 水口大知: SLP による組込みシステム要求仕様のレビュー Ver. 2.0, http://www.jfp.co.jp/slp/SLP_review_20100916.pdf(2010).
12) G. A. Miller: The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information, Psycol. Rev., Vol. 63(1956), pp. 81-97.
13) 宮崎和人，安達太郎，野田晴美，高梨信乃: 新日本語文法選書 4 モダリティ，くろしお出版 (2002).
14) 松吉俊，江口萌，佐尾ちとせ，村上浩司，乾健太郎，松本裕治. テキスト情報分析のための判断情報アノテーション. 電子情報通信学会論文誌 D, Vol. J93-D, No. 6, pp.705–713, 2010