

保全文書を対象とした故障分類可視化システムの開発

二神 廉太郎^{1,a)} 室 啓朗¹

受付日 2022年3月10日, 再受付日 2022年5月11日/2022年6月27日,

採録日 2022年8月8日

概要: 電力アセットなどの保全の現場では、作業内容を記録した保全文書を作成・蓄積し、ナレッジとして故障発生時における故障対策の立案業務等に活用している。しかしこれらの保全文書は、情報抽出が困難な非構造データであるため人手で1件ずつ精読する必要があり、活用の手間やコストが問題となっていた。そこで我々は保全文書を活用したソリューションとして、保全文書を機械的に解析して故障の現象と対策を抽出し、ツリー状に整理した故障分類木を利用者に提示することで、故障復旧の迅速化を支援する故障分類可視化システムを開発した。本システムでは、保全文書に固有の表現を抽出した辞書を用いることで、保全事例の抽出、現象と対策の関係性分析、故障分類木による可視化を行う。故障分類木は、検索クエリの故障現象に対して、過去の事例で同時に生じた故障現象とその対策を整理し、その分類を分岐で表現して可視化したものであり、大量の保全文書の内容を横断的かつ迅速に把握可能とする。加えて、故障分類木は保全文書に記載された現象と対策を抽出して機械的に作成可能であり、従来のFT図などの可視化と比較して低コスト化が可能である。本手法を用いて保全文書を可視化することで、保全文書の67%の内容を保ったまま14%のノード数に圧縮することを実験にて確認し、故障対策の立案業務の迅速化が可能な見通しを得た。

キーワード: 保全文書, 情報抽出, 故障分類木, 故障分類可視化システム

A Development of Fault Pattern Visualization System for the Maintenance Reports

RENTARO FUTAGAMI^{1,a)} KEIRO MURO¹

Received: March 10, 2022, Revised: May 11, 2022/June 27, 2022,

Accepted: August 8, 2022

Abstract: In the assets maintenance domain, maintenance reports itemizing work done are used as a source of knowledge in activities such as deciding how to repair the faults. Unfortunately, because they are made up of unstructured data that is difficult to analyze, making use of these records takes a lot of effort and cost, needing to be read through by a human one entry at a time. In this study, we have developed Fault Pattern Visualization System in which a system for representing fault classification details assets faster recovery from faults by using the machine analysis of maintenance reports to identify the observed phenomenon and the repair action. The information is presented to users in the form of a fault pattern tree. The system uses keyword dictionary used in maintenance reports, extracting instances of maintenance work, analyzing the relationships between the phenomenon and repair actions, and then showing the results as a fault pattern tree visualization. This enables information from a large volume of maintenance records to be reviewed rapidly and all together. The visualization can be created at a lower cost than such as the fault tree, by analyzing maintenance reports automatically. As a result of evaluation, the visualization by the Fault Pattern Tree shows that it can compress to less than one seventh sentence with preserving 67% contents compared to the maintenance reports.

Keywords: maintenance reports, information extraction, fault pattern tree, fault pattern visualization system

¹ 株式会社 日立製作所 研究開発グループ デジタルプラットフォームイノベーションセンター
Hitachi, Ltd. Research & Development Group. Digital Platform Innovation Center, Kokubunji, Tokyo 185-8601, Japan.

^{a)} rentaro.futagami.ub@hitachi.com

1. はじめに

電力アセットなどの保全の現場では、日々の業務におけ

る保全の履歴を記録した保全文書が蓄積されている。保全文書とは一般的に、故障の保全対応を担う作業者が、保全対象アセットにおいて実施した点検の内容や観測された故障現象を記載し、その対策として実施した作業の内容や交換した部品、対策後の経過などが記載される。

蓄積された保全文書は、保全の作業に対して点検内容や故障対策の指示を行う保全計画者の業務などに活用されている。保全計画者の業務では、保全対象アセットにて故障が発生したときに、その故障エラーコードやそのアセットで起きている表面的な故障現象の情報が伝えられると、保全文書を参照することで過去の保全事例と比較を行い、最適な点検や対策内容を計画して作業者に指示を行う。保全計画者には、保全文書を参照することで非属人的に最適な保全計画を立案することが求められるとともに、その計画を速やかに実施し保全対象アセットの復旧を迅速化することが求められる。そこで、保全文書の情報をデジタル技術を活用することで抽出・整理し、関連する保全事例の検索やその内容把握を迅速化する支援を行うことで、保全計画を迅速に実施することが期待される。

しかし、保全文書は情報抽出が困難な非構造データであり、蓄積したデータを活用して業務を効率化するためには、データから有用な情報を抽出し整理することが必要とされる。一般に、保全の最適化を支援する技術を導入するためには、そうした保全データを収集し利用可能とする作業に、導入工数全体の80%の工数が必要であると報告されている [1]。特に保全文書の情報抽出の問題となるのが、現場レベルの報告をベースに記述したため生じる、多数の専門用語や表記ゆれ、不統一なフォーマット、係り受けの曖昧な文章構造などであり、これらが汎用的な自然言語処理技術の適用を困難にする要因であった。そのため、保全文書を対象に情報抽出を行った従来研究では、検索した保全文書を精読するために時間を要することや、情報抽出に用いる故障知識を手で網羅的に定義するために工数を要するといった課題が残されていた [2], [3]。

筆者らは、保全計画業務の支援においては、故障の現象と対策を低コストで抽出しその情報を保全計画者が迅速に把握可能なよう、可視化を行う手法が有効であると考えた。それらの故障の現象と対策は、事前に列挙可能な故障の部位を表す語句および現象と対策を表す語句の組み合わせによって内容を理解可能であることに着目した。こうした語句の定義を行うための固有表現抽出や類義語検出の技術は、人手による少数の定義を基に機械学習を行い、文書全体に定義する手法がすでに確立されており [4]、少ない工数で利用することができる。

本研究では、固有表現や類義語の辞書を利用して保全文書からの情報抽出を行い、その結果を容易に把握可能な形式に可視化する手法を提案する。その可視化の方法として、一般的に用いられる FT 図 [5] のような故障の原因や

因果関係を表すものでなく、故障対策の立案に有用な故障の現象と対策の分類結果をツリー状に表現する、故障分類木と呼ぶ新しい形態を提案する。また、保全計画業務において把握している故障のエラーコードと現象をユーザが入力すると、対応する故障分類木を表示することで保全計画者が必要な対策を瞬時に把握することを支援する、故障分類可視化システムを開発した。本手法により作成した故障分類木を、専門家が作成した正解の故障分類木と比較することで、保全文書の情報を再現できた割合と、文章の数を削減し把握時間を短縮できた割合を評価した。

2. 従来研究と課題

2.1 従来研究

本節では、保全文書を対象として情報抽出を行った従来研究について述べる。

Farley ら [2] は航空機の保全ログブックから修理の対象機器と、修理のアクションを抽出した。抽出した結果は迅速な保全対応が必要なときに、ケースベースの検索に使用された。これによりユーザは航空機に生じた各問題に関連する文書を容易に取得することが可能になった。ただし、その検索結果として文書を整理した可視化などは提供されず、内容把握に必要な時間の短縮は研究対象とされなかった。

安藤ら [3] は、船舶機関の保全に必要なデータを故障報告書のテキストマイニング技術を用いた分析手法を検討した。具体的には、(1) 故障報告書からのイベントの抽出、(2) イベント間の関係の獲得、(3) 関係の評価という3つのプロセスによって、高い精度で文中で報告されているイベント、イベント間関係を取り出すことが可能となった。この技術を用いて抽出した結果は FT 図 [5] の形式に可視化され、故障の内容把握を容易化した。この研究では対象ドメインに関しての、各用語の同義語・分類階層構造・所属関係を定義したドメイン知識辞書をオントロジとして定義する必要があった。そのため情報抽出や可視化の実行に必要な工数の削減と、情報抽出の精度を両立することが課題であった。

Cadavid ら [6] は、生産ラインの保全ログデータから少ない準備で情報抽出を行う手法を検討した。具体的には、大量の自然言語を深層学習を用いて学習することで、文章から汎用的な特徴抽出が可能な BERT [7] モデルの多言語版である CamemBERT [8] を利用した。この研究では (1) CamemBERT を用いて保全ログデータから特徴を抽出し、(2) 各故障における重症度と故障の持続時間の予測を行う技術を提案することで、動的生産計画の立案支援を行った。ただし、各故障に対してどういった現象がみられ、どういった対策が取られたかといった自然言語的な情報抽出は研究対象とされなかった。

2.2 課題解決のための着眼点

1章で述べたように、保全計画業務においては保全対象アセットに生じた故障と関連する保全文書を抽出し、そこに記載された現象と対策を参考に最適な対策内容を短時間で決定することが求められる。この業務遂行を支援するためには、目的の保全文書の検索を効率化するだけでなく、保全文書の現象と対策を整理し可視化することで、複数の関連する保全文書の内容把握に必要な時間を短縮することが必要である。

ところで保全の業務としては、発生した個々の故障に対して即応的に対処し復旧することを目的とする保全計画業務のほかに、多くの機器において頻出する故障に対してその根本原因を分析し、その発生を抑制するための根本対策案や次期製品に向けた改良案を検討する品質保証業務がある。この品質保証業務の支援には、故障に関連した情報を可視化するための公知の手法である故障木 (FT 図) による可視化が有効である。FT 図は故障が発生した原因や因果関係を木構造で表現することを目的としており、木の根に配置された故障の表面的な事象からその原因になりうる事象をたどることで、故障の根本原因を分析することが可能である。

このように、FT 図は品質保障業務の支援に有効であるが、故障の因果関係を扱うためにその作成は保全のナレッジを必要とするなど難易度が高い。一般に保全文書には保全作業が行った作業の履歴が主に記載されており、根本原因やその因果関係が分かることは稀であり記載されない場合がほとんどであるため、保全文書を機械的に解釈して FT 図を構築することは困難である。そのため FT 図を作成するためには熟練の保全計画者のナレッジを言語化して作成する必要がある。作成に膨大なコストが必要となる。一方で、保全計画業務においては必ずしも故障の原因そのものを特定することは必要なく、発生した故障現象に対する対策の特定が必要なため、保全文書に記載された過去の事例を参照するだけで多くの故障に対処可能である。したがって保全計画業務の支援においては、現実的なコストで保全文書から機械的に作成可能な可視化として FT 図とは異なる手法が求められる。

本研究では、保全計画業務における過去事例の参照を効率化するために、保全文書から低コストで作成可能な可視化を研究対象とした。

本研究で目指すような、保全文書に記載された現象と対策の可視化が実現できたとして、その可視化から保全計画業務に必要な情報を網羅的に把握することと、その情報を少ない可視化要素から短時間で読み取ることが求められる。そのために、現象と対策の取捨選択と要約を行い、必要な情報を維持したまま可視化に表示する要素数を削減することで可読性を向上させる処理が必要である。なぜなら保全文書の記述は作業者の業務報告がベースとなっている

ために、実施した点検や観測した現象をその重要さ・必然性を考慮せず網羅的に記載する傾向があり、故障の特徴的な事象は何であったか判別することが課題となるためである。その判別が必要な例として、「A には異常がみられたが、B は正常であった。そのため X の交換を行った。」という故障事例を想定して説明する。この故障事例の潜在的な状況として、次の 2 種類のケースが想定できる。

(1) 作業指示書にある A の検査を行ったが、A の検査だけでは対策を決定できなかったため、追加で B の検査も行った。

(2) 作業指示書にある A と B の両方の検査を行った。

(1) のケースでは A と B の両方の状態が対策を決定するために重要である。一方 (2) のケースでは、A と B の故障は独立しており、A の異常と X の対策が関係を持つことになる。その場合、B が正常であるか検査した記述は X の交換とは直接関係がないため、保全計画者が計画立案の参考とする際には不要な記述であり、B の異常と関係する対策 Y が存在するときに限って必要とされる記述である。

保全文書中には (2) のケースが多くみられることもあり、故障分類の可視化を行った際のノイズとなって、可読性を大きく損なう要因となる。本稿において故障分類木の可読性が高いことは、保全計画業務に必要な情報を網羅するという制約の下で、故障分類木を構成する要素数が少なく短時間で把握可能であることを示す。保全文書の可視化を行う技術には、前述のノイズを除去するフィルタリング処理により可読性を向上させることで、迅速に情報を把握可能とすることが望まれる。このフィルタリング処理により、従来研究の課題であった、可視化に必要な工数とその内容把握の容易さを両立することが期待できる。本研究では故障の現象と対策を保全文書から低工数で抽出し、重要な情報を選択して可視化することで、故障復旧に必要な対策を短時間で把握することを支援する技術の提案を目的とする。

3. 故障分類可視化システム

本研究では、2.2 節で述べた課題を解決するために保全文書を可視化する、故障分類可視化システムを提案する。故障分類可視化システムは FT 図よりも保全計画業務の支援に適した、故障分類木と呼称する新しい可視化形態を用いる。本章では初めに故障分類木の概要を述べた後、故障分類可視化システムの構成を示し、その構築方法を述べる。

3.1 故障分類木

本節では、保全文書の現象と対策を迅速に把握可能な可視化形態である、故障分類木を提案しその特長を述べる。故障分類木とは、過去の保全履歴を記録した保全文書などを参考にして構築を行う、本研究で提案する可視化形態で

ある。その用途として、保全計画者が保全計画を立案する際に必要な情報を効率的に得るための活用を想定している。また、本節では一般的な故障現象の可視化手法として知られる故障木 (FT 図) と、故障分類木との違いについて述べる。

図 1 にある故障現象としてエラーコード 001 番が発報した場合に、過去に併発があった現象とそのときの対策を分類して示した故障分類木の例を示す。故障分類木は、図 1 のように任意の故障エラーコードや故障の現象などを根ノード (以下、クエリノード) として灰色で表現し、その子ノードには現象と対策をそれぞれ青色と赤色で表した、ツリー状の可視化形態である。故障分類木は故障の現象とその対策の組を分類し、分岐を用いて適切に整理を行った結果である。その故障分類は、根から葉までの一連の現象と対策の組で表現されており、少なくともその分類の故障が過去に 1 件発生していることを示している。保全計画を立案する際には、故障分類木の根から順に発生した現象をたどることで故障分類を特定し、その葉に示された適切な対策を容易に把握することができる。例として、根と右端の葉までの現象と対策の組からは、「エラーコード 001 が発報し、点検を行うと現象 E が観測されたため、対策 B の処置を行った。」といった情報を読み取ることができる。また、故障分類木のそれぞれのノードの大きさは、それぞれの現象や対策の発生頻度を表している。このノードの大きさを参照することで、発生確率の高い故障分類を優先的に確認でき、2 つの対策候補がある故障分類においてより主要な対策を選択することが可能になる。

故障分類木のクエリノードは、利用者が必要とする故障情報を絞り込む要素であれば、任意の事象・属性を指定可能である。そのため、図 1 のように故障現象を指定する以

外にも、別に定義された故障エラーコードなどを指定し、そのエラーコードを持つ故障事例を対象に故障分類木を作成することも可能である。また図 1 のように、各故障事例に付与されている故障分類などのメタ情報などを黄色のノードで表現して根の下の階層に挿入し、利用者が日常的に利用する分類方法を取り入れた形態も想定される。

保全対象の機器に故障が発生したときに、故障エラーコードや速報的に伝わった故障現象をクエリとして故障分類木を閲覧し、現象ノードを参照することで発生している見込みが高い現象を把握し、その発生を確認するために必要な点検内容を計画することが可能である。また、それぞれの現象ノードの葉にある対策ノードを参照することで、点検の結果現象が確認された場合に実施すべき対策の候補を把握し、計画に含めることが可能である。故障分類木に表示される内容は、後に示す整理処理によって、保全文書から抽出した情報のうち正常に抽出された重要度の高い現象と対策に絞って表示される。したがって、故障分類木を活用することで保全計画者の技量によらず適切な保守計画を短時間で立案することが可能である。

故障分類木による可視化は、一般的な故障現象の可視化手法である FT 図とは異なる方式であり、その効果においても違いがある。図 2 に FT 図の例を示す。図 2 にあるように、FT 図は表面的に観測できる現象を根とし、その現象の背後に根本的な原因が存在しており、その根本原因が AND または OR の関係で寄与している因果関係を表現した図である。また、2.2 節で述べたように、FT 図は保全文書の内容を抽出することで機械的に作成することは困難であり、熟練の保全計画者が手作業で作成することが必要となる。一方で、故障分類木は保全文書に記載された現象と対策を抽出することで機械的に作成することが可能であ

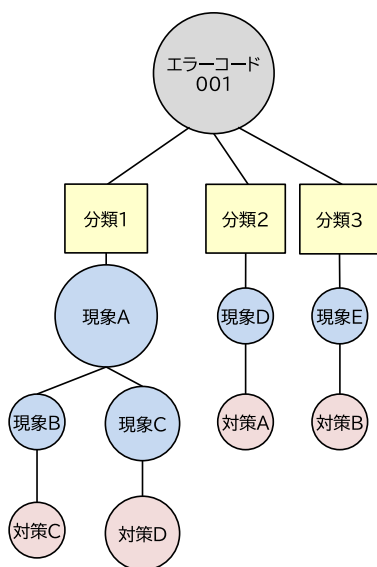


図 1 故障分類木の一例

Fig. 1 Example of fault pattern tree.

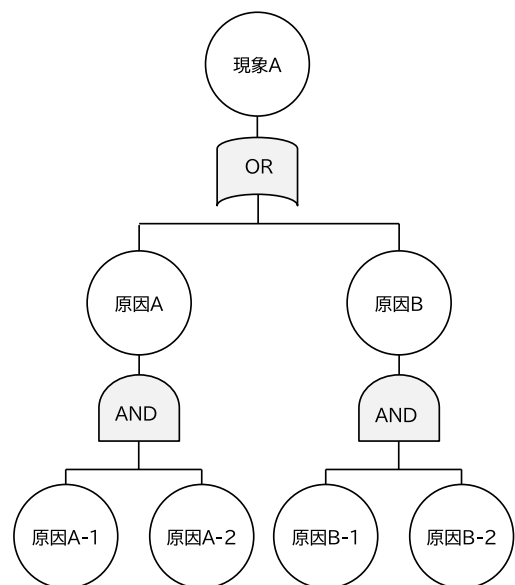


図 2 FT 図の一例

Fig. 2 Example of fault tree.

り、保全計画業務において必要なそれらの情報を重点的に把握可能である。あるいは、原因間の因果関係の抽出・表現を含まず保全文書に記載された原因を現象と対策に加えて表示したい場合においては、その原因を表す用語を定義することで故障分類木に表示することが可能である。

以上の理由のために、保全計画業務を支援する目的において、故障分類木は FT 図よりも適した可視化の形態といえる。また、2.1 節で述べた従来技術と比較して、故障分類木は保全文書に一般的に記載されている故障の現象や対策といった情報から作成することができ、複雑な現象の因果関係を推定することや、その処理に必要な事前知識の作成工数を削減できる点において優位性がある。具体的には、以降の節で述べる技術を用いると、保全文書とその固有表現を定義した辞書を入力することで機械的に故障分類木を出力することが可能である。加えて、故障分類木により故障の現象と対策の効率的な把握を支援することは、故障対策を直接提示することによる支援と比べて、対策実施のトリガーとなる現象を把握する効果があり、より適した対策を選択するうえで有効である。

3.2 故障分類可視化システムの全体構成

図 3 に故障分類可視化システムの全体構成を示す。故障分類可視化システムの構築に必要な入力データは、保全文書とその固有表現を定義したキーワード辞書である。

加えて、より可読性の高い故障分類木の出力を目指す場合には、固有表現の類義語を定義した類義語辞書の入力が可能である。これらの入力データを故障分類可視化システムに与えることで、保全文書に記載されている故障現象とその対策を抽出し、抽出された現象と対策を統合・整理する。これらの処理の結果を用いて故障分類木を生成し、図 3 の左下に示すシステム GUI に結果を表示する。システム GUI 上でユーザーが故障分類木のクエリノードを指定することで故障分類木の生成処理が逐次実行され、対応する故障分類木を表示することが可能である。故障分類可視化システムは、このシステム GUI を保全計画者に提供することで、故障発生時に適切な故障分類木を提示し、対策立案に必要な情報の把握を支援するシステムである。

以降の節では、3.3 節で入力するデータの形式や制約について、3.4 節では現象と対策の抽出処理について、3.5 節で現象と対策の整理処理について、3.6 節で故障分類木の出力処理と GUI の詳細についてそれぞれ述べる。

3.3 入力データの形式

本研究で対象する保全文書として、保全事例を識別する ID と、その保全事例における点検・作業内容を自然言語で記録した表 1 に示す例のようなデータを想定している。表 1 では report 列に点検・作業内容を格納しており、case id 列はその保全事例の ID である。report 列の条件として、故障分類木に表示されるべき故障の現象と対策が、その対象部位と観測あるいは作業した内容の順に記載される必要がある。report 列と case id 列は必須であり、加えてあらかじめ定義された故障分類に関するメタ情報を付与し、故障分類木による可視化に明示的に分類を加えることが可能である。たとえば、表 1 の case type 列のように保全対象機器のどの系統に関する故障かを記載した列や、保

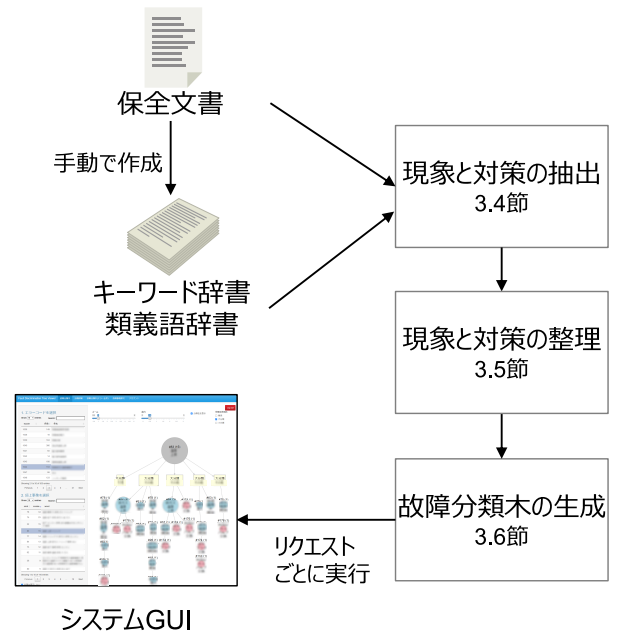


図 3 故障分類可視化システムの構成

Fig. 3 System architecture of fault pattern visualization system.

表 1 保全文書の例

Table 1 Example of maintenance reports.

| case id | case type | report |
|---------|-----------|--|
| 1 | 機械系 | 顧客から部位 A の起動時に現象 O1 発生との連絡あり。部位 B に現象 O2 があるとのこと。その場で部位 C を対策 A1 した。 |
| 2 | 電気系 | 部位 A の起動時の現象 O1 により部位 C から部位 D 現象 O3 が発生。以下の対策を実施。 ・ 部位 C の対策 A1 ・ 部位 A 内部の対策 A2 |
| 3 | 電気系 | 部位 A の起動時に現象 O1 と部位 B 現象 O2 が見られた。部位 C の現象 O4 が原因だと推定。部位 A 対策 A2 を行った。 |

全対象機器において発報されたアラームコードなどを格納した列などが想定される。保全計画者による評価から得られたプラクティスとして、故障分類木にこうしたメタ情報を表示すると、可読性は低下するが内容把握がしやすくなるため、表示の有無の切替ができるとよいと評価いただいた。

また、一般に想定する保全文書の中には、report 列に相当する箇所に「【状況】」や「【対策】」といった文書の構造化や、日付や日時、「・」や「①」などの箇条書き記号のような文書修飾が用いられる場合がある。本研究で提案する保全文書の解析では、こうした文章の規格・修飾を前提としない手法であるため、上記のような記号については前処理の段階で除去し、単純な文章形式に加工して report 列に格納が必要である。

次に、保全文書に含まれるドメイン固有の表現を定義したキーワード辞書について述べる。キーワード辞書は、ドメイン固有の表現を格納した word 列と、その語の持つ意味情報を格納した label 列を持つ。

先に示した保全文書を対象としたキーワード辞書の例を表 2 に示す。このキーワード辞書には、故障分類木で表現したい現象や対策を構成する単語をすべて列挙する必要がある。保全文書から効率的にキーワード辞書を作成する方法としては、一件ずつ保全文書を精査して重要な単語にアノテーションを行い、すでにアノテーション済みの単語は自動的にハイライト表示しながら、アノテーションされていない重要な単語がみられなくなるまで繰り返すことが有効である。

ここで label 列に記載するラベルの種類は自由に決めてよい。ただし、現象と対策のそれぞれを想起させるラベルは必須であり、この例では observation, approach としている。残りの保全対象機器の部位を表す target と故障現象の発生時期を表す situation といったラベルは自由に粒度を設定することができ、たとえば部位の細かさによって submodule など付与したり、故障エラーコードを表す

表 2 キーワード辞書の例

Table 2 Example of keyword dictionary.

| label | keyword |
|-------------|----------------------------|
| target | 部位 A, 部位 B, 部位 C, 部位 D |
| situation | 起動時 |
| observation | 現象 O1, 現象 O2, 現象 O3, 現象 O4 |
| approach | 対策 A1, 対策 A2 |

表 3 保全文書から抽出されたキーワードとそのラベルの例

Table 3 Example of keywords and labels extracted maintenance reports.

| | | | | | |
|-------------------------------|-----------|----------------------|--------|---------------------|----------|
| 顧客から部位 A の起動時に現象 O1 発生との連絡あり。 | | 部位 B に現象 O2 があるとのこと。 | | その場で部位 C を対策 A1 した。 | |
| 部位 A | 起動時 | 現象 O1 | 部位 B | 現象 O2 | 部位 C |
| target | situation | observation | target | observation | target |
| | | | | | 対策 A1 |
| | | | | | approach |

error などといったラベルを追加したりできる。後の処理でラベルを内部的に統合したり、無効化したりが可能のため、キーワード辞書を作成する段階で厳密にラベルセットを検討する必要はない。また、保全文書には「〇〇は異常がなかった」などのように現象を否定する表現が頻出する。この否定を表す「なかった」などもキーワードとみなし、negative ラベルなどを付与することが必須である。総じて、target, observation, approach, negative のラベルが必須である。ここで故障の原因も同様に抽出したい場合には、その原因を示す用語をたとえば cause ラベルを付与するなどして定義する必要がある。

また、保全文書ではその記載に表記ゆれが多く含まれており、それらを整理しない場合は故障分類木の現象や対策が冗長に表示される恐れがある。そのため、類義語辞書を用いた類義語置換処理が有効である。この類義語辞書として、たとえば「冷却水」という代表語に対して「ジャケット水」や「ジャケット冷却水」を類義語として登録する。

3.4 現象と対策の抽出処理

前節で定義したキーワード辞書を用いて、保全文書の意味をまとまりごとに抽出して、故障分類木のノードの単位で格納する処理を行う。このとき各ノードに対して、そのノードで表現する事象は現象または対策のどちらを表しているのか機械的に判定する処理を行う。その抽出結果として、各保全文書の ID と、その保全文書から抽出されたキーワードおよびそのラベルと、それぞれのキーワードに付与されたノード ID およびそのノード種類の判定結果を得る。

その抽出のために、まずは保全文書から順序を保持したままキーワードを抽出する。先に示した「顧客から部位 A の起動時に現象 O1 発生との連絡あり。部位 B に現象 O2 があるとのこと。その場で部位 C を対策 A1 した。」という保全文書の文章例について、キーワード辞書に定義された表現とマッチングした結果を表 3 に示す。

表 3 に示した例では、一貫して target や situation の後に observation や approach が来ていることが分かる。ここで、前者のラベルを目的語ラベルと呼び、後者のラベルを述語ラベル (negative ラベルも含める) と呼ぶ。この目的語ラベル、述語ラベルのセットごとをノードとして定義する。すなわち、述語ラベルの後に目的語ラベルが現れる場合、そこでノードを分割する。また、読点や記号などでも適宜分割を行う。このとき、上記のノード分割のルール

に従うと、1つのノードに複数の目的語ラベルおよび述語ラベルを含む場合があるが、以降の処理において問題はない。

次に、分割したノードの構文チェックを行い、解釈不可能なノードを削除する処理を行う。先に述べたように、自然な文章からノードを抽出する場合、ノードには目的語ラベルと述語ラベルのキーワードがそれぞれ1つ以上含まれるべきである。表2のラベル定義においては、target ラベルのキーワードが1つ以上あり、observation ラベルまたは approach ラベルのキーワードが1つ以上あることが必須である。そうでないノードは、解釈が困難な文章構造であるため、削除する。また、目的語ラベルの situation ラベルと、述語ラベルの negative ラベルは、その語単体では現象または対策が成り立たないため、単体で現れた場合は同様に削除する。ここで述べた解釈不可能なノードは、キーワード辞書に定義した固有表現に抜けがある場合や、元々の文書が表3のような典型的な表記でなく、「下記の交換を実施した。部位 A。部位 B」のように倒置的な表記であった場合に生じる。

そして、それぞれのノードごとに現象または対策を表すかを特定し、ノード種類を付与する。そのためにノードに含まれる述語ラベルの種類を参照し、ルールベースでノード種類を決定する。例として、あるノードに含まれる述語ラベルが observation のみであるとき、そのノードを現象ノードと判断し、approach を含む場合は (observation を含む場合でも) 対策ノードと判断するなどが適当である。4章の実験においては、上記のルールに従いノード種類を付与する処理を実施した。

表4に本節の処理によって保全文書から抽出した現象と対策の抽出例を示す。ただし node id 列は抽出されたノードの ID であり、type 列はノード種類である。

3.5 現象と対策の整理処理

前節で抽出したノードは、表記ゆれであったり、点検で正常が確認されたすべての箇所だったり、保全文書の冗長と思われる記述も網羅的に抽出されている。こうした冗

表4 保全文書からの現象と対策の抽出例

Table 4 Example of phenomenon and action extracted maintenance reports.

| case id | node id | type | label | keyword |
|---------|---------|------|-------------|---------|
| 1 | 1 | 現象 | target | 部位 A |
| 1 | 1 | 現象 | situation | 起動時 |
| 1 | 1 | 現象 | observation | 現象 O1 |
| 1 | 2 | 現象 | target | 部位 B |
| 1 | 2 | 現象 | observation | 現象 O2 |
| 1 | 3 | 対策 | target | 部位 C |
| 1 | 3 | 対策 | approach | 対策 A1 |
| ... | ... | ... | ... | ... |

長な記述は、後に故障分類木による可視化を行った際に、可読性を低下させる要因となる。そこで、1. 類義語辞書によるキーワード置換 (以下、類義語置換)、2. ノード間の類似度に基づく統合 (以下、ノード統合)、3. 重要でない現象のフィルタリング (以下、現象フィルタリング)、4. 故障分類に現象が無い場合にその対策を削除 (以下、孤立ノード削除) といった4種類の整理処理を行うことで故障分類木の可読性を向上させる。

これらの整理処理を行うことで、2.2節にて示した2種類のケースにあるようなノイズを除去する効果が期待できる。その場合、(1)のケースの場合にはAとBの両方の現象が対策の立案に寄与したと判定され、故障分類木において両方の現象が表示されるが、(2)のケースの場合にはただ点検しただけで対策立案に寄与しないBの現象はノイズと判定され故障分類木にはAの現象のみが表示される。この判定は3.5.3節にて示す現象フィルタリング処理により行われ、文書中のAとBの出現順などによらず、他の故障事例において現象Aのみを観測して対策を立案できた事例を発見することで実現する。こうした判定を機械的に行う方式では、特に現象フィルタリング処理において発生件数の少ない現象に対してその重要度が誤判定されることにより、重要な情報が誤って削除される場合もある。しかし、保全の現場から得られたプラクティスとして、類義語置換処理とノード統合処理のみを実施した状態では内容が冗長で見にくいとのご意見があった。その対策として、本研究では現象フィルタリング処理および孤立ノード削除処理の追加を提案したところ、出力された故障分類木が正解故障分類木とほぼ一致する結果になったとの評価をいただいている。

3.5.1 類義語置換処理

本処理では、抽出結果の各キーワードが類義語辞書に定義された類義語と一致するとき、そのキーワードを代表語に置換する処理を行う。

その例として、「冷却水の温度」や「ジャケット水の温度」のように実際には同じ冷却水を示す複数の表現がみられる。4章の実験においては、3.3節で述べた類義語辞書を用いて、「ジャケット水」などの表現を代表語である「冷却水」に置換する処理を行った。本処理の実施後も、類義語辞書では定義しきれない表現方法の差異が残る場合が多く、十分な可読性をまだ確保できていない。

3.5.2 ノード統合処理

本処理では更なる可読性の向上のため、任意の2つのノードのキーワード群の類似度を計算し、その類似度が閾値以上である場合はそれらのノードを類似ノードとして同じノードIDを付与する処理を行う。本処理によって類義語置換処理では整理しきれない表記揺れを吸収することが可能である。類似度の閾値の決定は、後に示す処理によって故障分類木に可視化した際に適切な粒度で整理されるよ

う適宜調節する。

ノード間の類似度の計算は、ノードに含まれる特定のラベルのキーワードすべてについてコサイン類似度を計算し、その類似度が最小になる組み合わせの和を計算することで行う。ここで特定のラベルとは、目的語ラベルあるいは述語ラベルに該当する1件のラベルである。たとえば現象ノードの類似度計算の場合は、target ラベルと observation ラベルをそれぞれ特定のラベルとして計算する。また、対策ノードの場合は、target ラベルと approach ラベルのそれぞれを特定のラベルとして計算する。

比較する対象のノード A とノード B の特定のラベル n についての類似度 $D_n(W_{An}, W_{Bn})$ は (1) 式により計算する。

$$D_n(W_{An}, W_{Bn}) = \frac{1}{|W_{An}| + |W_{Bn}|} \left(\sum_{a \in W_{An}} \min_{b \in W_{Bn}} d(a, b) + \sum_{b \in W_{Bn}} \min_{a \in W_{An}} d(b, a) \right) \quad (1)$$

ただし、 W_{An} はノード A に含まれる特定のラベルが n であるキーワードの集合であり、 a は W_{An} の元である。 d はコサイン類似度関数である。また、ノード A とノード B の類似度はすべての特定のラベル n についての $D_n(W_{An}, W_{Bn})$ の平均値である。ただし、negative ラベルの有無によって現象の肯定と否定を判断するため、ノード A とノード B で negative のキーワードの有無が異なるときノード間の類似度は 0 とする。

3.5.3 現象フィルタリング処理

本処理は、2.2 節で述べた保全文書に含まれる現象のノイズを除去するための処理である。その処理内容として、ある故障事例において各現象ノードが対策を決定するためにどれだけ重要であるか計算し、その重要度が閾値以下であった現象ノードを削除する。現象ノードの重要度は、機械学習を用いて現象ノードから対策ノードを予測するタスクを行い、そのタスクにおける各現象ノードの変数重要度を XAI (eXplainable AI) 技術を用いて計算する。予測を行うための機械学習アルゴリズムとしては決定木 [9] やランダムフォレスト [10] などが有効であり、その変数重要度として、Gini 係数の減少量 [10] や SHAP [11] などの手法が適当である。

4 章の実験においては、ランダムフォレストによる予測と Gini 係数の減少量を使用した。これはランダムフォレストによる予測が精度と計算速度の両面において優れており、Gini 係数の減少量が計算速度の面で優れているためである。こうした計算速度に優れた手法を選択することで、現象と対策の整理処理にかかる時間を短縮し、故障分類可視化システムの GUI のレスポンスを改善可能である。

3.5.4 孤立ノード削除処理

本処理は、ある故障分類を特定するために十分な現象が

記述されておらず対策のみが記述されている故障分類において、その対策を削除する処理である。たとえば、故障分類木に極めて表層的で多くの故障分類で共通して生じる現象（以下、頻出現象ノード）が現れ、その直下に故障を特徴づける他の現象なしで対策が現れる場合がある。これは表記揺れなどの要因で現象ノードの抽出に失敗し対策ノードのみが抽出されることが原因であり、頻出現象ノードのみでは故障分類として意味をなさないため対策ノードを削除する必要がある。

3.4 節で述べた現象と対策の抽出処理では、ノードの内容自体が解釈不可能な場合にそのノードを削除する処理を行った。対して、本処理では現象ノードの抽出に失敗し対策ノードのみが孤立して得られた場合に、その対策ノードを削除する。

具体的な頻出現象ノードの基準としては、故障分類木に記述するすべての現象の中で、各現象の出現頻度が q パーセント以上であることとする。この q の値としては 99 を推奨し、この値を用いたとき、故障分類木の全現象の出現頻度の分布のうち 99% を超える出現頻度の現象が親ノードである対策ノードを削除する処理となる。ただし、表示する現象の数が少ない小規模な故障分類木の場合、多頻度の現象であっても 99% の閾値を超えないため、99~95% 程度の範囲でパラメータとして調節するとよい。

3.6 故障分類木の出力処理

保全文書から抽出した現象と対策を、故障の分類が把握できるよう故障分類木として可視化する処理について述べる。このとき、保全計画者が必要な現象と対策の表示に絞り込み可読性を向上させるために、まずはクエリノードを指定する必要がある。このクエリノードとしては、故障事例に付与されたエラーコードや故障分類コードなどを指定してもよい。故障発生時のディスパッチとして故障分類木を用いる場合は、クエリノードは故障機器に表示されたエラーコードや第一に観測された表面的な現象などが想定される。

次に、3.4 節で述べた保全文書からの現象と対策の抽出結果の中から、クエリノードと同じ事例で発生した（エラーコードが与えられた場合は該当するエラーコードの事例の）現象と対策のノードを抽出し、その結果を子として配置する。そのとき、現象ノードを配置した下に対策ノードを配置し、それぞれの並び順としてはより多くの事例で記載された現象または対策を浅い階層に配置する。ノード配置の過程で同じ（ノード統合された）現象が同じ階層に現れた場合は1つのノードにまとめて表示し、その子を分岐構造として表現する。ただし、可読性を高めるためには一度分岐した子が下の階層で合流することは避け、木構造を維持することが望ましい。同様に、違う階層に同じノードが配置された場合であっても、統合・合流は行わない。

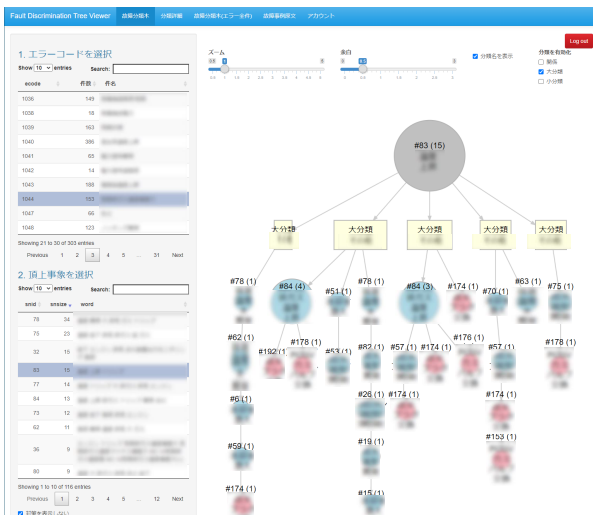


図 4 故障分類可視化システムの GUI
Fig. 4 GUI of fault pattern visualization system.

この抽出の事前に、3.5 節で述べた類義語置換処理とノード統合処理が行われているものとする。また、抽出後に現象フィルタリング処理と孤立ノード削除処理を行うものとする。

故障分類可視化システムでは、図 4 に示した故障分類可視化システムの GUI を通じてユーザが指定するクエリノードを対話的に受けつけ、上記の処理によって逐次適切な故障分類木を出力し、ユーザに提示することで保全業務を支援する。クエリノードの入力方法として、図 4 のサイドバー上段にあるテーブルからエラーコードを指定し、その結果を受けて下段のテーブルに絞り込み表示された関連する故障現象をクエリノードとして選択する。この操作によって指定したクエリノードを基に構成された故障分類木が画面右側に表示される。

4. 結果の検討および効果

保全対象アセットとしてガスエンジンの保全を行う事業者が所有する、ガスエンジン保全文書を対象に 3 章で述べた故障分類可視化システムを適用した。その結果を基に故障分類木による可視化の技術的な評価と、その可視化を行うために必要な工数について述べる。

4.1 評価方法と条件

本研究で提案した故障分類木の定量的な評価の方法と、その評価指標である対策再現率と情報圧縮率、作成時間について述べる。

故障分類可視化システムは情報の抽出・整理を行い故障分類木として可視化を行う技術である。この技術の評価するために、保全事業者が保有するガスエンジン保全文書と、その保全事業者にて作成いただいた故障分類木の正解サンプル（正解故障分類木）を提供いただき、提案手法を用いて出力した故障分類木と比較した。ガスエンジン保全

文書は表形式のデータであり、表 1 と同様な構成に加えて各故障事例のエラーコードが付与されている。

提案手法を用いての故障分類木の出力においては、913 件のキーワードとそのラベルを格納したキーワード辞書と、14 件の代表語とその類義語で構成される類義語辞書を手動で作成し、それぞれノード抽出処理と類義語置換処理に用いた。キーワード辞書の作成のために、ガスエンジン保全文書中に総じて 304 件あるエラーコードの中から無作為に抽出した約 400 件の保全文書に対し、文書中のキーワードを手で走査し辞書に追加した。本作業はキーワード辞書にすでに追加したキーワードをハイライト表示するツールを使用のうえ、ガスエンジン保全の非専門家が実施し、約 8 時間で完了した。なお、本作業の前半では文書から網羅的にキーワードを見つけるために時間を要したが、作業の後半は新規のキーワードがみられることは少なく短時間で完了した。従って本キーワード辞書作成におけるプラクティスとして、ガスエンジン保全文書の場合は 200 件程度の文書から網羅的にキーワードを抽出できれば、故障分類木の出力において実用的であると考えられる。また、図 1 の「分類 1」のようなメタ情報の付与は評価の対象外とし、故障分類木に表示せず評価した。その他パラメータとして、ノード統合処理の類似度の閾値を 0.4 とし、現象フィルタリング処理の重要度の閾値を 0.5 とし、孤立ノード削除処理の出現頻度の閾値を 99 パーセントとした。

正解故障分類木は、ガスエンジン保全文書のある 1 件のエラーコードに関する故障分類木であり、ベテラン保全計画者のドメイン知識を用いて事象の洗い出し・整理を行いつつ、その作成に十分な時間（約 20 時間）を費やし作成されたものである。そのため故障分類可視化システムの性能指標としては、出力される故障分類木が正解故障分類木の現象・対策をどれだけ再現できたか、どれだけ見やすく（短時間で把握できるように）整理できたか、といった観点で定量的に評価した。

注意点として、正解故障分類木は保全計画者の暗黙知も用いて作成されており、正解故障分類木に記載されている現象・対策が保全文書に記載されていない（他のエラーコードに分類されているなど）場合があった。そのため、故障分類可視化システムの可視化結果では、先に述べた性能指標がどれだけ保全文書から改善したかの観点でも評価した。なお、現象フィルタリング処理と孤立ノード削除処理の効果を見るために、故障分類木の評価は上記の整理処理を行わずクエリノードによる絞り込みの事前に行う類義語置換処理とノード統合処理のみを行った場合（事前整理のみ）と、上記処理を含めたすべての整理処理を行った場合（全整理あり）両方を対象に行った。その他に、故障分類木の作業時間は 304 件のエラーコードを対象とした結果であり、正解故障分類木の作業時間は 1 件のみのエラーコードを対象とした結果となっている点に注意されたい。

具体的な評価指標として、正解故障分類木に記載された根から葉までの現象と対策の組 20 件を再現できた件数の割合を対策再現率として計算した。また、現象と対策の削減と整理によって、内容を表示するために必要なノード数を削減した割合を情報圧縮率として計算した。情報圧縮率は故障分類木のノード数を、対象のエラーコードが付与された故障事例の文書の数で割った数とした。さらに、故障分類木の出力に必要なキーワード辞書の作成に要した時間、または正解故障分類木の作成に要した時間を作成時間として計測した。これらの故障分類木の性能指標の基準として、対策再現率は保全文書に近い数値を、情報圧縮率は正解故障分類木に近い数値を、作成時間はより少ない数値を示すほど有用だと考えられる。ただし情報圧縮率は、後に述べるように正解故障分類木にはない故障分類を故障分類木から見出すことを期待するため、完全に同一の数値を目指すわけではない。これらの評価指標は、保全計画業務に必要な情報を網羅するという制約の下で、故障分類木を構成するノード数が少なく短時間で内容把握できるか評価するため、故障分類木の可読性の高さを示す。

4.2 評価結果

表 5 に故障分類木の評価指標として、対策再現率、情報圧縮率およびその作成に要した時間を示す。

表 5 の対策再現率を見ると、事前整理のみの故障分類木は 70% の再現率、すなわち正解故障分類木に記載がある 20 組のうち 14 組を再現できたことが分かる。また、全整理ありの場合は 2 組が重要でないと判断されたことで対策再現率は 60% に低下した。さらに、故障分類木の生成の元になる保全文書でも、18 組 (90%) しか正解故障分類木を再現できなかった。これは正解故障分類木には記載がある一部の故障分類が、保全文書中では評価対象であるエラーコード以外に分類されていたことが原因である。これらの故障の影響を取り除き、保全文書の対策再現率を基準として故障分類木の対策再現率を補正すると、全整理ありでは 67%、事前整理のみでは 78% となった。

前記補正があっても対策再現率が 100% にならない理由として、ノード抽出の不足や整理処理の過剰などがある。

このうち現象フィルタリング処理では、発生頻度が高くサンプル数が多い現象と対策においては重要度を正確に計算できるため再現されやすい。したがって、故障分類木は発生頻度の高い故障の多くに対応可能であり、対応可能な故障事例の割合を考えると対策再現率を上回ると予想できる。また、今回の実験においては現場の保全計画者の意見を参考に情報圧縮率を重視するよう整理処理のパラメータを設定した。そのため低頻度の現象と対策の表示を優先する場合は、このパラメータの設定を変更することで対策再現率を向上させることが可能である。

次に、表 5 の情報圧縮率を見ると、事前整理のみの故障分類木では 127 の現象と 78 の対策を用いて分類を表現しており、全整理ありの場合は 52 の現象と 50 の対策となった。これは 751 文の文章が記載された保全文書と比較して、文章数を現象数と対策数の和と考えると、全整理ありの場合では 27%、事前整理のみの場合は 14% にまで圧縮された。また、正解故障分類木と比較しても故障分類木は全整理のなし・ありでそれぞれ 4 倍、2 倍程度の文章量に抑えられた。参考として、故障分類木可視化システムの抽出処理により 751 文の保全文書から抽出された現象・対策ノードの数は計 730 ノードあり、3.4 節で述べた解釈不可能なノードを取り除くと計 690 ノードが抽出された。これに 3.5 節の 4 種類の整理処理を用いることで 102 ノードにまで削減し可視化できたため、把握に必要な時間を大きく削減できたといえる。

図 5、図 6 に正解故障分類木と同じエラーコードが付与された保全文書を可視化した故障分類木を示す。図 5 は事前整理のみで、図 6 は全整理ありの場合の結果である。ただし、部位、現象、対策を表すそれぞれのキーワードについて、部位 A などのようにダミー表示している。これを見ると、全整理ありの場合の図 6 の故障分類木は、8 種類程度の主要な故障分類があることを示しており、それぞれの詳細と対策を把握しやすい可視化になっている。一方図 5 の故障分類木は、故障分類が非常に多岐にわたるとともに、同じ現象が複数の故障分類にみられるなど冗長な結果となっており、現象フィルタリング処理と孤立ノード削除処理の必要性を示している。以上の結果から、故障分類木

表 5 故障分類木の評価結果

Table 5 Result of fault pattern tree evaluation.

| | 保全文書 | 故障分類木 (事前整理のみ) | 故障分類木 (全整理あり) | 正解故障分類木 |
|------------------------|-------|-------------------|------------------|---------|
| 対策再現率 (保全文書比の対策再現率) | 90% | 70% (78%) | 60% (67%) | 100% |
| 情報圧縮率 | 100% | 27% | 14% | 7% |
| 現象ノード数 | | 127 | 52 | 29 |
| 対策ノード数 | | 78 | 50 | 21 |
| 合計ノード数 (または文章の数) | (751) | 205 | 102 | 50 |
| 作業時間 | 0 時間 | 8 時間 | 8 時間 | 20 時間 |

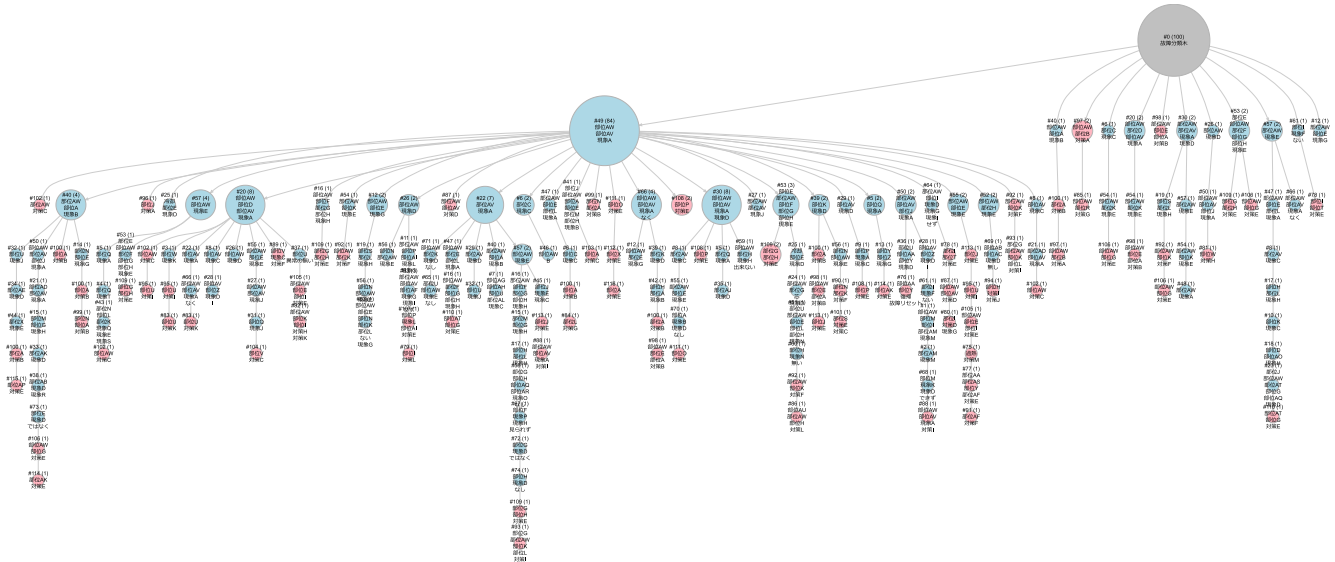


図 5 現象フィルタリング処理と孤立ノード削除処理を適用していない故障分類木の例

Fig. 5 Example of fault pattern tree without executing phenomenon filtering and isolation deletion.

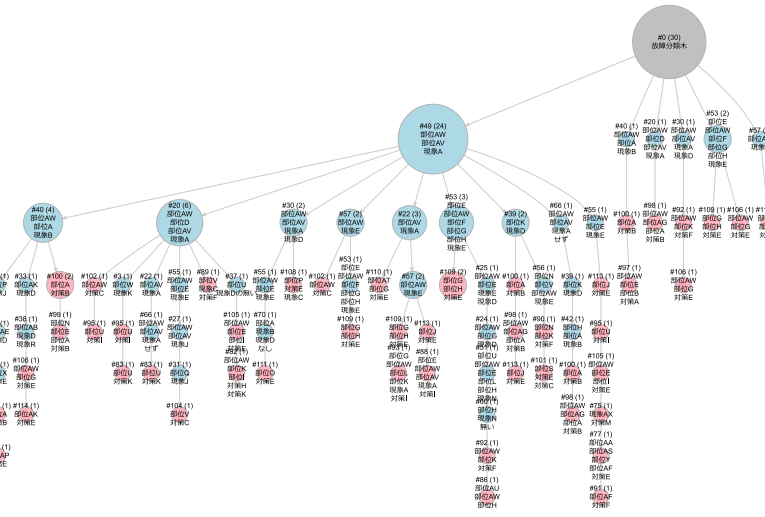


図 6 現象フィルタリング処理と孤立ノード削除処理を適用した故障分類木の例

Fig. 6 Example of fault pattern tree with executing phenomenon filtering and isolation deletion.

の構成にはすべての整理処理を用いることが望ましい。

次に、表5の作業時間を見ると故障分類木の出力には、キーワード辞書の作成のために8時間の作業を要したのに対して、正解故障分類木の作業時間は20時間であった。この結果から単純に比較すると、故障分類木の出力は正解故障分類木の作成に比べて半以下の作業時間で作成可能であった。これは提案手法がキーワード辞書を作成するだけで、現象と対策の抽出および整理を自動的にを行い、作成時間を短縮したためである。さらには、この結果は正解故障分類木が1件のエラーコードのみが対象であるのに対して、故障分類木は304件すべてのエラーコードを対象とした作業時間であった。したがって、すべてのエラーコードを対象とした正解故障分類木と比較する場合、エラーコードの件数と比例してさらなる作業時間の短縮が期待できる。

なお、故障分類可視化システムをガスエンジン保全文書に適用した際の処理時間としては、ノード統合処理の際に行う類似度計算がほぼすべてを占めた。故障分類木表示の事前に行う現象と対策の抽出処理と事前整理処理の実行時間の目安は、CPU: Inter Xeon 5600 番台 150 Cores, MEM: 950 GB の計算機において、およそ7万件のノードに対して約4時間を要した。その他の現象フィルタリング処理や故障分類木の出力処理については、故障分類木の出力要求に応じて逐次処理を行っても十分に短い時間で完了でき、たかだか1秒程度で完了した。

ただし、これらの処理時間は表5の作成工数に含まない。仮に故障分類木の出力に必要な8時間の作業時間に上記の4時間の計算時間を加えたとしても、合計12時間で304件のエラーコードすべてに故障分類木を出力することができ、これはエラーコード1件あたり0.04時間であるた

め、手作業による正解故障分類木の作成に 20 時間要することと比較して 500 分の 1 に短縮可能であった。

評価の結果、保全文書に故障分類可視化システムを適用し故障分類木を閲覧することで、保全文書の 67% の情報を保ったまま文章量を 14% に削減（事前処理のみの場合は 78% の情報を 27% の文章量に削減）することが可能なが示された。故障分類木はこのような大幅に少ない文章量でもって、保全文書の内容をツリー状に整理した結果であるため、保全計画時に過去事例の内容把握に必要な時間を 7 分の 1 以下に削減可能であると試算できる。

また、保全現場における上記システムの試行から得られたプラクティスとして、全整理ありの故障分類木であれば、若手の保全計画者の業務の支援としては十分実用的であるとの評価のほか、故障発生時の第一報を受け付けるコールセンターに導入すれば簡単なディスプレイパッチを実施するために活用できると評価をいただいた。

本節の実験において、故障分類木の出力およびキーワード辞書の作成には適用対象のドメイン知識および専門家の作業を要していないため、提案手法は適用先のドメインを限定しない汎用性があると考えられる。

以上の結果から、故障分類可視化システムの有用性が示された。

5. おわりに

本研究では、保全計画業務の実施を支援し保全対象アセットの復旧を迅速化することを目的とする故障分類可視化システムを開発した。また上記支援のために、保全文書に記載された故障現象と対策をツリー状に整理した新しい可視化形態として故障分類木を提案し、その可視化を機械的に出力する方法を提案した。

故障分類木の評価実験において、保全文書から機械的に出力した故障分類木と専門家が作成した正解故障分類木を比較したところ、故障分類木は故障分類の 67% を表現できており、その文章量を 14% に削減することを確認した。これにより故障対策を立案するために行う過去の故障事例の把握に必要な時間を 7 分の 1 に削減可能な見通しを得た。

故障分類木の作成に必要な作業時間として、提案手法を用いることで非専門家の 8 時間程度の作業を要するのみで 304 件ものエラーコードの故障分類木を機械的に作成でき、専門家が手作業で作成すると 1 件あたり 20 時間が必要であったのに対し、大幅な省力化が可能であることを確認した。また本提案手法は適用対象のドメインおよびその専門知識に依存しておらず、汎用性が高い特長がある。

今後の課題として、本研究では故障分類木の有効性を定量的に示したが、故障分類木による保全計画業務の工数削減効果についても同様に評価する必要がある。また、故障分類木による可視化において、故障分類を漏れなく表現す

るといった精度の向上も技術的な課題といえる。その具体的な対策としては、保全文書に仮定した単純な係り受けを改め、保全文書の実態に即した例外的な処理を導入することが考えられる。

謝辞 研究の方向付けならびに技術課題の検討において、株式会社日立パワーソリューションズの藤田 聡志氏、小村 昭義氏および株式会社日立製作所の伊藤 政昭氏、Hidayah, Noor 氏に大変なご協力をいただいたことを深謝する。また、保全文書のご提供ならびに評価において、株式会社日立パワーソリューションズの平山 究氏と渡邊 浩二氏に大変なご協力をいただいたことを深謝する。

参考文献

- [1] 新井信行：予兆検知システムとデータおよび分析駆動型「戦略的メンテナンス」、火力原子力発電, Vol.7, No.2, pp.186-190 (2019).
- [2] Farley, B.: Extracting information from free-text aircraft repair notes, *AI EDAM*, Vol.15, pp.295-305 (2001).
- [3] 安藤英幸ほか：テキストマイニングを用いた故障報告書分析手法の研究, 日本造船学会論文集, Vol.192, pp.475-483 (2002).
- [4] Zhang, C.: DeepDive: A Data Management System for Automatic Knowledge Base Construction, *Univ. of Wisconsin-Madison Thesis*, (2015).
- [5] Lee, W. S., et al.: Fault Tree Analysis, Methods, and Applications — A Review, *IEEE trans. on reliability*, Vol.34, No.3 pp.194-203 (1985).
- [6] Cadavid, J. P. et al.: Valuing free-form text data from maintenance logs through transfer learning with CamemBERT, *Enterprise Information System*, (2002).
- [7] Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, *arXiv e-prints*, arXiv: 1810.04805, (2018).
- [8] Martin, L. et al, CamemBERT: a Tasty French Language Model, *arXiv e-prints*, arXiv: 1911.03894, (2019).
- [9] Safavian, S. R. and Landgrebe, D.: A survey of decision tree classifier methodology, *IEEE Transactions on Systems*, Vol.21, Issue 3, pp.660-674, (1991).
- [10] Breiman, L.: Random forests, *Machine Learning*, Vol.45, No.1, pp.5-32 (2001).
- [11] Scott, M. L., and Su-In L: A Unified Approach to Interpreting Model Predictions, *Advances in Neural Information Processing Systems*, (2017).



二神 廉太郎

2018 年筑波大学大学院システム情報工学研究科博士前期課程修了。2019 年同大学大学院システム情報工学研究科博士後期課程退学。同年株式会社日立製作所入社。現在、エンタープライズ向けプラットフォーム技術の研究開発に従事。



室 啓朗

1993年大阪大学大学院電子工学専攻修士課程修了。同年株式会社日立製作所入所。現在、製造プラント、社会インフラネットワーク等におけるデジタルツイン向けデータモデルの研究開発に従事。