

定性的信頼性／ 安全性解析支援ツールの開発

土肥 正[※]岡村 寛之[※]

本研究は、FTA (Fault Tree Analysis), FMEA (Failure Mode and Effect Analysis), HAZOP (Hazard and Operability Study) などの定性的信頼性/安全性分析を支援するためのツール開発を行う。具体的に、設計や障害事例などに関する過去の情報を非構造型データとしてデータベースに蓄積し、FTA, FMEA, HAZOPに現れる故障モードやガイドワードなどのキーワードと、対象とするシステムの設計情報 (UML / SysML) を手がかりに、関連する過去の事例を抽出する枠組みを提案する。更に、それらを重要度に従ってランキングする。これは安全性分析者が設計段階における工数削減に寄与する。

Development of a Support Tool for Qualitative Reliability / Safety Analysis

Tadashi Dohi[※], and Hiroyuki Okamura[※]

In this article, we develop a support tool for qualitative reliability/safety analysis such as FTA (Fault Tree Analysis), FMEA (Failure Mode and Effect Analysis) and HAZOP (Hazard and Operability Study). The developed tool contains the database for unstructured data; UML/SysML diagrams and prior incidents, and picks up the related prior events from given failure modes of FTA and FMEA, guidewords of HAZOP and UML/SysML diagrams. Based on these events, our tool also provides the ranking of events with their severities. This would be helpful for safety analysts to reduce their efforts in the phase of system design.

1 はじめに

ソフトウェア内の欠陥がシステムの安全性を脅かす事例は枚挙に暇がない。ソフトウェアの欠陥が発生する要因は幾つか存在するが、設計時における「考慮漏れによる欠陥」は、フィールドにおいて最も深刻な障害を引き起こす可能性のある欠陥であることが多い。同時に、この種の欠陥は最も取り除くのが困難な欠陥の一つとして知られている。一般的なシステム設計では、システムが配置される外部環境において発生し得る事象を網羅的に列挙し、その事象が起きた場合にシステムがどのようなレスポンスを示すかを分析する。これは、単に外部環境要因とシ

ステム間で相互に関連する事象を列挙するだけでなく、システム内における特定のコンポーネントが故障して初めて外部環境に伝搬されるような、システム内部のコンポーネント間における交互作用効果も注意深く分析する必要がある。このように、考慮漏れによる欠陥は常にヒューマンエラーによって引き起こされる欠陥であり、現実の開発期間内で完全に防ぐことは極めて困難である。

考慮漏れによる欠陥を防止するためには、すべての事象を網羅的に分析することが重要となる。とくに、安全性を阻害する故障はこれまでに未経験であることが多く、事象の発生をあらかじめ想定することが難しい。安全性分析は「想定外を想定する」

※ 広島大学

という難しい課題に挑戦しなければならない。具体的な安全性分析手法として、FTA (Fault Tree Analysis), FMEA (Failure Mode and Effect Analysis), HAZOP (Hazard and Operability Studies) などがある。FTAでは、ある発生してはならない障害(例えば、人命にかかわる事故など)が発生する要因や事象をトップダウン的に洗い出す手法であり、最終的には、故障木と呼ばれる事象と原因の関係を記述するグラフモデルを構築する。FMEAは、FTAとは逆にボトムアップ的な手法であり、あるコンポーネントで典型的な故障(故障モード)が発生した場合に起こる影響を積み重ねることにより、システム全体にどのような障害が起きるかを分析する。HAZOPはシナリオに基づいた分析手法である。その考え方は故障モードを用いたFMEAによく似ており、故障モードの代わりにガイドワードと呼ばれるプロセスに対する異常な操作や状態を用いて、プロセス異常がシステムに与える影響を分析する。これらは、信頼性や安全性を数値的に定量化するのではなく、信頼性や安全性を損なう要因を定性的に分析することから、定性的信頼性/安全性分析手法と呼ばれる。

実務レベルで、これらの定性的信頼性/安全性分析を行う場合、二つの大きな問題がある。一つ目は、分析結果が分析者の経験に大きく依存するという問題である。FTA, FMEA, HAZOPは、本来、経験したことのない事象であっても考慮漏れによる欠陥が入らないような設計を行うための手法であるが、システムに関する知識が豊富な分析者と経験が少ない分析者では、結果として作成された分析結果が大きく異なる。FTA, FMEA, HAZOPは本質的に、事象の原因、故障モードとその影響、ガイドワード

などをきっかけとして、分析者への連想を促す手法と言える。つまり、分析者の知識や想像力などの個人的な能力による所が非常に大きいため、最終的な分析結果の品質に対するばらつきが大きく、「経験の少ない分析者が不十分な分析を行う」というリスクが依然として残る。二つ目は分析に要するコストの問題である。FMEAやHAZOPはボトムアップ的な分析手法であり、すべてのコンポーネントに対象となるすべての故障モードを適用する、あるいは、すべてのプロセスに対してすべてのガイドワードを適用することが推奨される。しかしながら、その組み合わせはコンポーネント数の増加やプロセスの複雑化によって指数関数的に増加する。先にも述べたように、定性的信頼性/安全性分析は、そもそも分析者の連想に依存した試行錯誤的なマニュアル分析であるため、爆発的に増加する組み合わせをすべて網羅するためには膨大な人的コストがかかる。そのため実務では、分析者の経験則から「最も危険なコンポーネントやプロセス」を経験的に特定し、その部分の信頼性・安全性を集中的に分析することが多い。この点においても、分析者の経験不足による考慮漏れのリスクが常に存在する。

本論文では、自動車制御のような高い信頼性/安全性が必要なシステム開発において、このような問題点を部分的に解決若しくは緩和する目的で、定性的信頼性/安全性分析を支援する汎用的な技術の開発と定性的信頼性/安全性分析を支援するツールの開発を行う。具体的に、過去の情報(設計やHAZOPによる安全性分析結果など)を蓄積し、対象とするシステムの設計情報(UML / SysML)で注意すべきコンポーネント(プロセス)

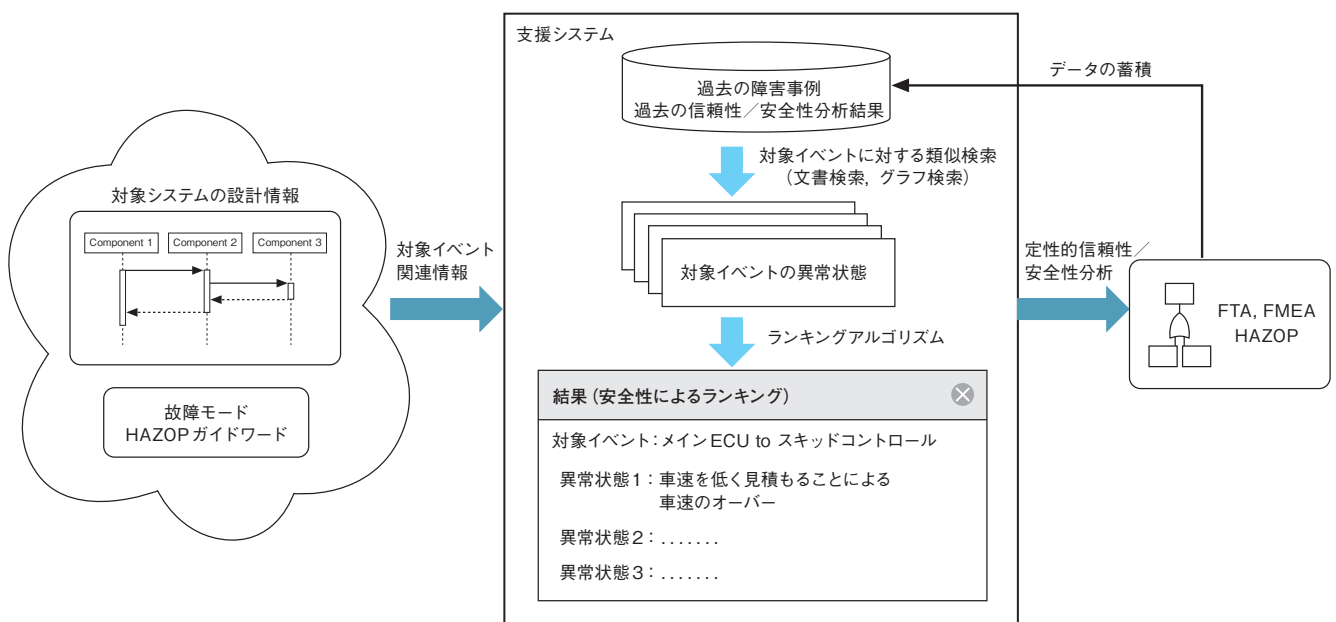


図1 ツールの概要

を重要度に従ってランキングするための学習アルゴリズムを開発する。とくに、定性的信頼性/安全性分析はシステム設計の早い段階で行う必要があるため、ソフトウェア開発の上位レベルでの定性的信頼性/安全性分析を支援することを目的とする。

2 ツールの概要

本論文では、対象システムの設計情報と過去の情報(安全性分析結果)から対象システムの設計上で重要なコンポーネント(プロセス)を推定し、優先順位を付けて分析者に提示する支援ツールの開発を目標とする。支援ツールは過去のHAZOP分析結果と現在の設計の類似度を評価し、その類似度に基づいた重要

度スコアリングを行う(図1参照)。そのため、過去の分析結果を学習するフェーズと現在の設計を評価(スコアリング)するフェーズに分けられる。図2及び図3は作成する支援ツールのユーザースペースを示している。図2は過去の分析結果を学習するフェーズであり、(1)一般的な文章からの単語(タグ)の抽出及びベクトル化、(2)定性的信頼性/安全性分析に関連した文章からのタグの抽出及びベクトル化、(3)過去の設計書及び分析例の分解と登録されている単語による特徴付け(タグ付け)を行う。図3は現在の設計書の入力により評価を行うフェーズを表しており、(1)設計書の分解及びタグ付け、(2)タグに基づいた過去の分析例との類似検索、(3)類似度と重要度のスコアリングを行う。

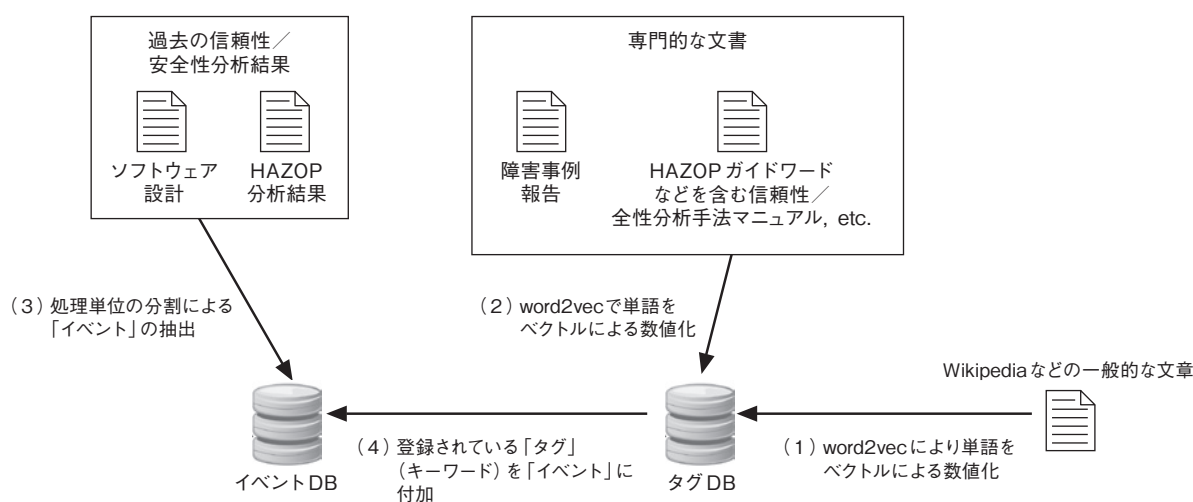


図2 学習フェーズ

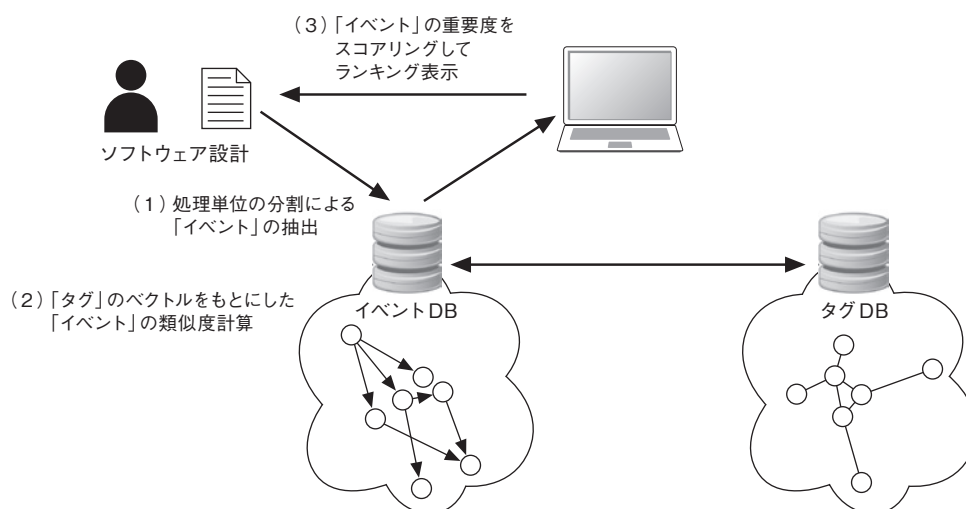


図3 評価フェーズ

3 類似度検索手法

検索手法の調査・開発では文章データとUMLによる図で記述された設計書を统一的に扱う必要があるため、データ構造の定義が重要になる。安全性分析結果とソフトウェアの設計情報という、一見異なるものの類似性(関連性)を評価する必要がある。つまり、一般のドキュメント間での類似検索技術を開発する必要がある。

データ構造については、設計書や安全性分析を構成する「文章」と、その「文章」を構成する「単語」に着目したデータ構造を定義することで、すべてのデータを统一的に扱う仕組みを構築する。また、類似性の評価についても「単語」に着目した類似性の定義を行う。

3.1 データ構造の定義

テキスト文書及びUMLシーケンス図を格納するデータ構造として図4に示すクラスを用いる。テキスト文書及びUMLシーケンスは表1の単位の「文章」に分解されEventクラスを生成する。

Eventクラスは、識別子(id)、名前(name)、対応するテキスト文書(description)、重要度(score)の属性を持つ。また、イベント間の関連を表すためのリンク(link)属性も持つ。更に、descriptionをもとに「単語」の抽出が行われTagクラスを生成する。Eventクラスに関連するTagクラスはそれぞれ文書内の単語の出現頻度に応じた重み(weight)を持つ。Tagクラスは、識別子(id)、キーワード(keyword)と後述するword2vec^[1]によって算出されたベクトル(vector)の属性を有する。descriptionから日本語の単語を抽出するためにMeCab^[2]による「わかち書き」を適用した。

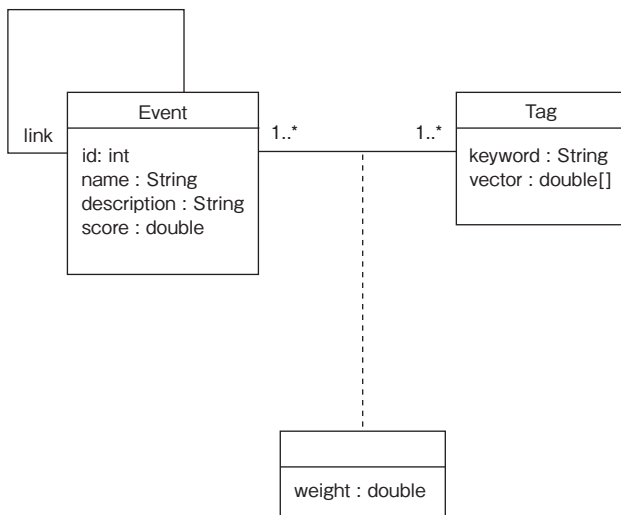


図4 データ構造のクラス図

表1 Eventクラスの単位

ドキュメント	イベント単位
テキスト文章	パラグラフ
UMLシーケンス図	メッセージ、オブジェクト、コメントなどUMLの構成要素単位
HAZOP分析	一つのガイドワードに対する分析結果

3.2 イベント間類似度の定義

Eventクラス間の類似度を算出するために、本論文ではタグの数値ベクトル化を行う。j番目のEventクラスの数値ベクトル v_j は、Tagクラスの数値ベクトル w_1, w_2, \dots (word2vecによって得られる)とEventクラスに対するTagクラスのTF-IDF値をもとに以下の式で得られる。

$$v_j = \sum_i \text{tfidf}_{ij} w_i.$$

ここで、 tfidf_{ij} はTagクラスiのEventクラスjにおけるTF-IDF値である。一般に、ある単語に対するTF-IDF値は文書中出现する単語の頻度(TF: Term Frequency)と、その単語を含む文書の頻度(DF: Document Frequency)から算出され、DFについては逆数の対数を取ったIDF(Inverse Document Frequency)を利用する^[3]。また、 w_i はTagクラスiのベクトルを表す。数値ベクトル化されたイベントの類似度に対しては、以下の式で定義されるコサイン類似度^[4]を用いる。

$$\text{sim}(A, B) = \frac{(\sum_{a \in S_A^n} a) \cdot (\sum_{b \in S_B^n} b)}{\|(\sum_{a \in S_A^n} a)\| \|(\sum_{b \in S_B^n} b)\|}$$

ここで、 S_A^n, S_B^n はEventクラスA及びBを中心にnホップ先までのイベントを含めた集合(a, bはそれらの要素(イベント)に対する数値ベクトルを表す)を表す。これはグラフ構造のデータに対する類似度(カーネル)定義^[5]を利用している。

3.3 Tagクラスのベクトル化

Tagクラスに対してはword2vecによるベクトル化を行う。word2vecはニューラルネットワークによって単語をベクトル化する手法を実装したツールである。具体的にはCBOW(Continuous Bag-of-Words)と呼ばれる図5に示すアーキテクチャを持つニューラルネットワークによって、自然言語で記述された個々の単語のベクトル化を実現する。

CBOWでは周辺の単語から適切な単語を予測する。図5の $w(t-2), w(t-1), w(t+1), w(t+2)$ は周辺の単語を表し、それらから目標とする単語 $w(t)$ を予測するようにINPUTとPROJECTION

間の重みを調整(学習)する. word2vecツールでは, わかち書きされた日本語を入力(コーパス)として, 単語をベクトル化する.

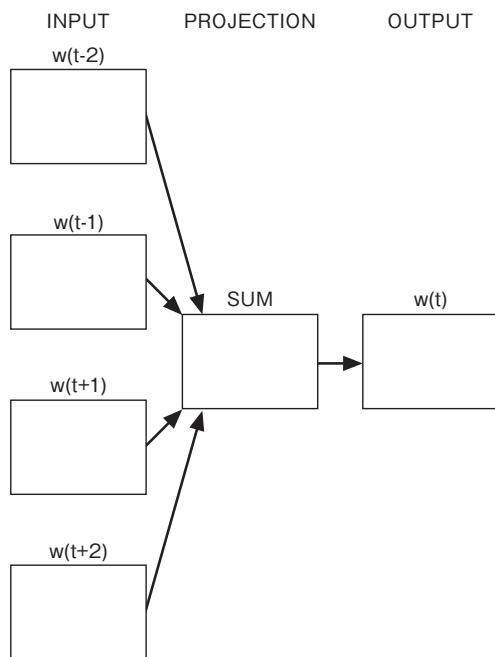


図5 CBOWのアーキテクチャ

3.4 重要度スコアリング

重要度の算出は, 文書(イベント)間の類似度と過去のHAZOP分析結果にあらかじめ付加されている重要度(HAZOPスコア)から行われる.

以下の記号を定義する.

$score(i)$: UMLシーケンスのイベント i の重要度

$sim(i, k)$: UMLシーケンスのイベント i とHAZOP分析結果 k の類似度

$score(k)$: HAZOP分析結果 k の重要度 (HAZOPスコア)

このとき, UMLシーケンスのイベント i に対して次の重要度スコアを算出する.

類似度が閾値 θ 以上のHAZOPスコアの平均

$$score(i) = \frac{\sum_k I(sim(i, k) > \theta) score(k)}{\sum_k I(sim(i, k) > \theta)}$$

ここで $I(\cdot)$ は指示関数を示す.

また, 上記を報酬としたマルコフ決定過程(MDP: Markov decision process)を用いたスコアリングアルゴリズム^[6]を用いる. MDPとは, 状態で行うアクションによって, マルコフ連鎖の推移確率が変化する確率過程を表し, 通常はマルコフ連鎖上で報酬構造を定義し, 報酬を最大にするアクションを行う.

今, ユーザがリンクをたどってノードを巡回するモデルを考える. Ω をノード全体の集合, ノード i のリンク先を Ω_i とする. MDPに基づいた枠組みでは, ユーザは各ノードのコンテンツの評価値に依存した最適な行動(移動)を行うものと仮定する. すなわち, ユーザがノード i を訪問しているときに選択できる行動を

$$A_i := \{\Omega_i, rnd\}$$

とする. ここで, Ω_i はリンクをたどって移動するノード, rnd はランダムなジャンプで別のノードへ移動することを意味する. ユーザは, これらの中で最も高い評価値が得られる行動を選択するものと仮定する.

ノード i を訪問することによって得られる報酬(重要度スコア)を r_i とする. 更に, 割引率 $0 < \beta < 1$ を定義して, n 回目の移動によって得られる報酬に β^n を乗じて価値を減じるものとする. この場合, より少ないノード移動でより多くの報酬を得ることが最適な行動となる. このMDPに基づいた枠組みを利用することにより, 単純にHAZOP分析結果との類似度だけで評価する場合よりも, UMLシーケンス上のリンク情報を加味した上での評価が可能となる.

具体的にMDPで各ノードをスコアリングするためには, 最適な行動を求める必要がある. 今, 次の記号を定義する.

$W^a(i)$: ノード i においてノード a への移動を選択し, 以後, 最適な移動を選択し続けた場合の総期待割引報酬.

$V(i)$: ページ i における最大総期待割引報酬.

これらを用いると, 最適性の原理^[7]から, 以下の最適性方程式を得る.

$$V(i) = \max_{a \in A_i} W^a(i),$$

$$W^a(i) = \begin{cases} r_i + \beta V(a), & a \in \Omega_i, \\ r_i + \frac{\beta}{|\Omega|} \sum_{i \in \Omega} V(i), & a = rnd. \end{cases}$$

上記の最適性方程式を満足する $V(\cdot)$ 及び $W(\cdot)$ を求めることで、総期待割引報酬を最大にするユーザの行動が決定される。最終的に、最適な行動が決定された後に、各ノードのスコアとして各ページに対する最大総期待割引報酬 $V(i)$ を用いる。またこれは、ページランクアルゴリズム [8, 9] の理論的な拡張にもなっている。

4 ツールの実装

開発したツールはGUIによるフロントエンド(図6参照)とデータベースから成る。GUIの作成にはJavaを用いた。開発したツールは、word2vec ツールによる出力をツール内のDBに登録する「タグDB登録」、過去の設計、過去のHAZOP分析結果、現在のUMLシーケンス図をDBに登録する「イベントDB登録」、登録されたイベントと類似するイベントを検索する「類似検索」、過去のHAZOP分析結果をもとに現在のシーケンス図のスコアリングを行う「重要度スコアリング」などの機能を有する。

「タグDB登録」ではword2vecから出力されたファイル(テキ

ストファイル)を読み込むことで、ツール独自の形式(バイナリ形式)に変換してタグを登録する。

「イベントDB登録」ではXML化されたドキュメントまたはドロツールから出力されるUMLシーケンス図のXMLファイルから、イベントのDB登録を行う。HAZOP分析結果をはじめとするドキュメント類は表計算ソフトウェアからXMLを作成して登録することができる。

「類似度検索」では、登録してあるイベント間の類似度に基づいた検索を行う。ここでは、コサイン類似度(COS similarity)及び関連するイベントを考慮したコサイン類似度(COS similarity with tags of one-hop neighbors)の2種類のイベント間類似度を用いた類似検索が実現できる。

「重要度スコアリング」は開発したツールの主要な機能であり、対象となるUMLシーケンス図とHAZOP分析結果を指定することで、シーケンス図上の各イベントに対する重要度を算出する。スコアリング手法には類似度を閾値として用いるものとMDPによるものをそれぞれ適用することが可能である。

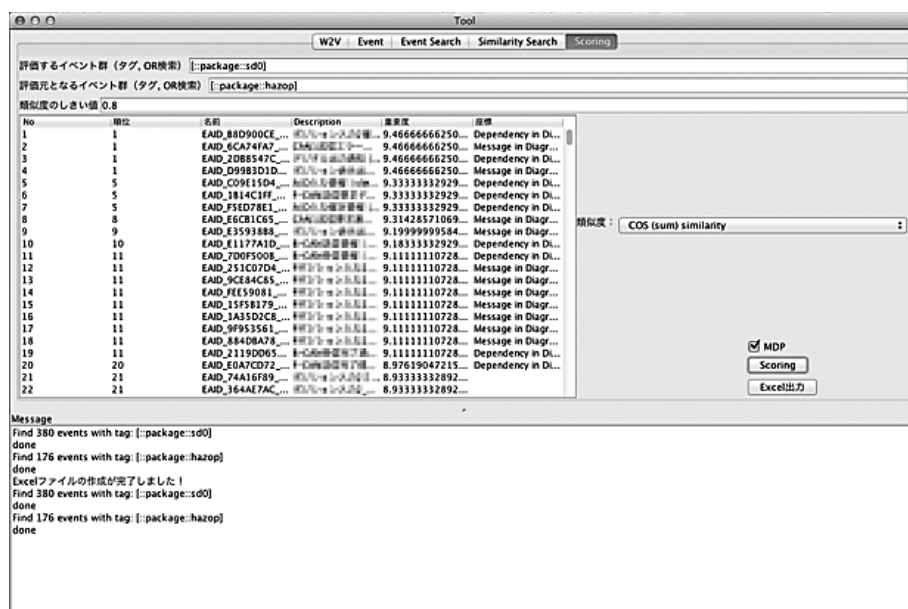


図6 ツールの実行画面(重要度スコアリング)

5 有効性検証

実際の企業(A社, B社)における設計者及び安全性分析者を対象に, 作成したツールの評価を行った. 実験においてはTagのデータを生成するため, Wikipedia, IPAが発行する電子書籍及び国際会議において収集した単語や文節データのテキストを利用し, 667,644単語に対するベクトル化を行った.

5.1 実証分析(A社)

A社で利用した設計データは, 自動車のシフト制御に関するシーケンス図であり, 2通りの異なる設計に基づいたシーケンス図(SD0, SD1)に対する重要度評価を行ったものである. 重要度スコアリングに必要となる過去のHAZOP分析結果は, A社で実際に同じアプリケーションドメイン(シフト制御)に対してHAZOP分析を行った結果(HAZOP1)と, JasPar(Japan Automotive Software Platform and Architecture)によって提供された自動車のトルク制御に関する安全性分析結果(HAZOP2)を用いて比較を行った. HAZOP1はシフト制御について安全性分析した結果であり, ポジション入力, CAN制御, デジタル出力における各機能, データ, ポートなどに対してHAZOPガイドワードを適用した障害シナリオを作成し, 安全性の検証を行っている. この分析結果に対して, 安全性分析者の手であらかじめ各障害シナリオの重要度を5段階(最も危険な障害を5)で評価してもらっていた. HAZOP2はJasParによる安全性分析例におけるHAZOP分析結果である. JasParではトルクセンサにおける安全性分析を行っており, 同様に上記のHAZOPガイドワードを適用して障害シナリオを生成している. 重要度についてはASIL(Automotive Safety Integrity Level)に応じた5段階の評価を行った. ASILでは, リスク分析を行った結果に応じて高いレベルの安全方策が求められる機能からD, C, B, A, QMの段階評価を付ける. 通常, これらの評価結果は過去の類似プロジェクトで既に分析したものを利用することが想定される.

HAZOP1を使った評価では, シーケンスSD0, SD1共に類似度の閾値を0.8と設定して評価を行った. SD0, SD1共にシーケンス中に全380イベントが存在したが, それぞれのシーケンスで100件程度のイベントを分析対象から外することができるという結果が得られた.

一方, JasParの結果(HAZOP2)を利用した場合, アプリケーションドメインが異なるため, 出現する用語が大きく異なり, 全体的に類似度が低下した. そのため, HAZOP1と比較すると有効なスコアリング結果が得られず, もととなるHAZOP分析と

シーケンスが同じアプリケーションドメインあるいは非常に近いアプリケーションドメインである必要があることが示された.

5.2 実証分析(B社)

B社で利用した設計データは, 空調制御に関するシーケンス図であり, このシーケンスに対する重要度評価を行った. 重要度スコアリングのもととなるHAZOP分析結果は同一ドメインのものが得られなかったため, JasParによって提供された自動車のトルク制御に関する安全性分析結果(HAZOP2)を用いて比較を行った. シーケンス図をイベントに分解して登録した場合のイベント数は24イベントであった. HAZOP2はASILによって重要度が与えられている.

HAZOP2を使った評価では, 先に述べたA社の分析と同様にアプリケーションドメインが異なるため, 閾値0.65と低く設定した. ツールによる評価では「EEPROM」, 「指定ブロック書込」, 「EEPROMミラーバッファ」及びその周辺のプロセスである「データ読出」, 「Verifyコマンド送信」, 「Disableコマンド送信」, 「Writeコマンド送信」, 「Enableコマンド送信」が比較的高い重要度として評価された. しかしながら, 実際のヒアリングでは, 「Backup Manager」などの項目が重要であるという指摘があり, ランキング精度についての問題が指摘された. その主な原因として, A社の場合と同様に異なるアプリケーションドメインのHAZOPスコアを用いたことが挙げられる.

6 まとめ

本論文では, 安全性分析において重要なコンポーネントを過去の安全性分析結果(HAZOP分析結果)から推定することによって, 安全性分析の工数削減を行うための支援ツールを開発した. とくに, 分析結果を構成する単語に注目して, 単語の頻度から文章間の類似度を算出する手法について考察し, それを用いた重要度分析手法について検討を行った.

過去において同様の研究はこれまでになされておらず本論文で行う試みは非常に挑戦的なものであった. 一方で, 通常の文書間における類似度分析とは異なり, ソフトウェア設計に使われる文章では, その文書量が圧倒的に少ない. 例えば, シーケンス図上のメッセージなどは短い文書で説明されていることが多く, その点を考慮した類似度評価を行う必要があったため, 本論文では単語そのものをベクトル化する word2vecと呼ばれる技術に注目し, それを利用した類似度評価と類似検索アルゴリズムを実装した.

類似検索についてはある程度の精度で実行できるが、実際的な問題としては、専門的な単語の登録など、辞書の整備をすることでより推定精度が高まることが期待される。一方で、現在の実装では、対象となるイベントの組み合わせを総当たりで類似度計算しているため、より多くの文書を扱う場合には計算を高速化するための工夫が必要になると考えられる。

重要度のスコアリングについては、分析対象のシーケンス図におけるすべてのイベントと過去のすべてのHAZOP分析結果の類似度を算出し、それらがある一定の類似度以上であれば、そのHAZOP分析結果にあらかじめ付加されている重要度を算術平均することでシーケンス上のイベントに対する重要度を評価している。実際、重要度を付けることによって、ある程度、分析に必要なイベントを特定することができ、安全性分析の工数削減が行える可能性があることは検証できた。しかしながら、実用にあたっては推定精度が十分ではなく、辞書の精緻化などによって類似度を高めることや重要度の評価精度を高める必要があることが分かった。また、過去のHAZOP分析結果とシーケンス図が同一のアプリケーションドメインである必要があるため「同一のドメイン」かどうかを判断するための指標を算出することも必要であることが分かった。

本論文では、HAZOP分析結果を前提として、シーケンス図とHAZOP分析結果の類似度を算出することで重要度を評価したが、今後の可能性としては、過去のシーケンス図上で重要な部

分があらかじめスコアリングされている場合には、対象シーケンスと過去のシーケンスの類似度を算出することで、対象シーケンスの重要度を評価する可能性も考えられる。

ツールそのものについては、プロトタイプであるため、入力及び出力がまだユーザにとって十分使いやすいものにはなっていない。とくに、出力については重要なコンポーネントがシーケンス図上のどこにあるかという情報を視覚的に明示することで、重要部分の特定を支援するための仕組みが必要である。また、入力については、HAZOP分析結果などの文書からの入力が特定のXMLによって与えられており、これを生成するための労力が必要である点が指摘できる。これらの問題については、統合開発環境との融合により解決すべき課題であると考えられる。更に、word2vecによるデータの管理や、イベント、タグの管理も現状はあまりスケーラブルな設計にはなっておらず、多くのデータから分析を行う場合など、より現実的開発状況においては分散処理などを行うことを検討する必要がある。

謝辞

本研究は独立行政法人情報処理推進機構技術本部ソフトウェア高信頼化センター（SEC：Software Reliability Enhancement Center）が実施した「2015年度ソフトウェア工学分野の先導的研究支援事業」の支援のもと行われたものである。

【参考文献】

- [1] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv: 1301.3781, 2013.
- [2] T. Kudo, K. Yamamoto, Y. Matsumoto, Applying conditional random fields to Japanese morphological analysis, Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pp230-237, 2004.
- [3] G. Salton, A. Wong and C.S. Yang, A vector space model for automatic indexing, Communications of the ACM, 18, pp613-620, 1975.
- [4] P.N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining: Pearson New International Edition, Pearson Education, 2014.
- [5] H. Kashima, K. Tsuda and A. Inokuchi, Marginalized kernels for labeled graphs, Proceedings of the 20th International Conference on Machine Learning, pp321-328, 2003.
- [6] 岡村 寛之, 宮内 聡, 土肥 正, マルコフ決定過程によるWebページランキングアルゴリズム, 電子情報通信学会論文誌 D-I, J89-D(2), pp210-219, 2006.
- [7] M. Puterman, Markov Decision Processes, John Wiley & Sons, New York, 1994.
- [8] S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine, Computer Networks and ISDN Systems, 30, pp107-117, 1998.
- [9] L. Page, S. Brin, R. Motwani and T. Winograd, The PageRank citation ranking: bringing order to the Web, Stanford Digital Library Technologies, Working Paper 1999-0120, 1998.