

2015 年度ソフトウェア工学分野の先導的研究支援事業  
「データマイニング手法を応用した定性的信頼性／安全性  
解析支援ツールの開発」  
成果報告書

平成 28 年 2 月

広島大学

本報告書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」の公募による採択を受け国立大学法人広島大学大学院工学研究院情報部門（研究責任者 土肥正）が実施した研究の成果をとりまとめたものである。

## 目次

研究成果概要 .....	1
1 研究の目的・背景と期待される効果.....	12
1.1 研究目的とその背景 .....	12
1.2 期待される効果 .....	14
2 実施内容.....	15
2.1 研究アプローチ .....	15
2.1.1 研究の全体像.....	15
2.1.2 関連するこれまでの研究について.....	16
2.1.3 研究目標と研究課題.....	17
2.2 研究の活動実績・経緯 .....	20
2.3 研究実施体制 .....	24
3 研究成果.....	28
3.1 研究課題1「検索手法の調査・開発」 .....	28
3.1.1 当初の想定.....	28
3.1.2 研究プロセスと成果.....	28
3.1.3 発生した問題および今後の展望.....	34
3.2 研究課題2「スコアリング手法の調査・開発」 .....	35
3.2.1 当初の想定.....	35
3.2.2 研究プロセスと成果.....	35
3.2.3 発生した問題および今後の展望.....	41
3.3 研究課題3「分析例の蓄積」 .....	43
3.3.1 当初の想定.....	43
3.3.2 研究プロセスと成果.....	43
3.3.3 発生した問題および今後の展望.....	49
3.4 研究課題4「ツールの開発・実装」 .....	50
3.4.1 当初の想定.....	50
3.4.2 開発プロセスと成果.....	50
3.4.3 発生した問題および今後の展望.....	64
3.5 研究課題5「有効性検証」 .....	66
3.5.1 当初の想定.....	66
3.5.2 研究プロセスと成果.....	66
3.5.3 発生した問題および今後の展望.....	70
4 考察.....	72
4.1 研究による効果や問題点等 .....	72
4.2 産業界への展開と今後の研究の進め方.....	73
4.2.1 研究成果の産業界への展開.....	73
4.2.2 今後の研究の進め方.....	73
4.2.3 産業界への要望.....	73

参考文献 ..... 75

## 研究成果概要

### 1. 背景

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いくつか存在するが、設計時における障害事象の「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。特に、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、事象の発生を予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」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、人命に関わる事故など）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 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値に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いえる。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ところ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、「経験値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を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。

本研究では、このような問題点を部分的に解決も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新しい技術の開発と開発現場において分析を支援するツールの開発を行う（図1参照）。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）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。特に、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、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。

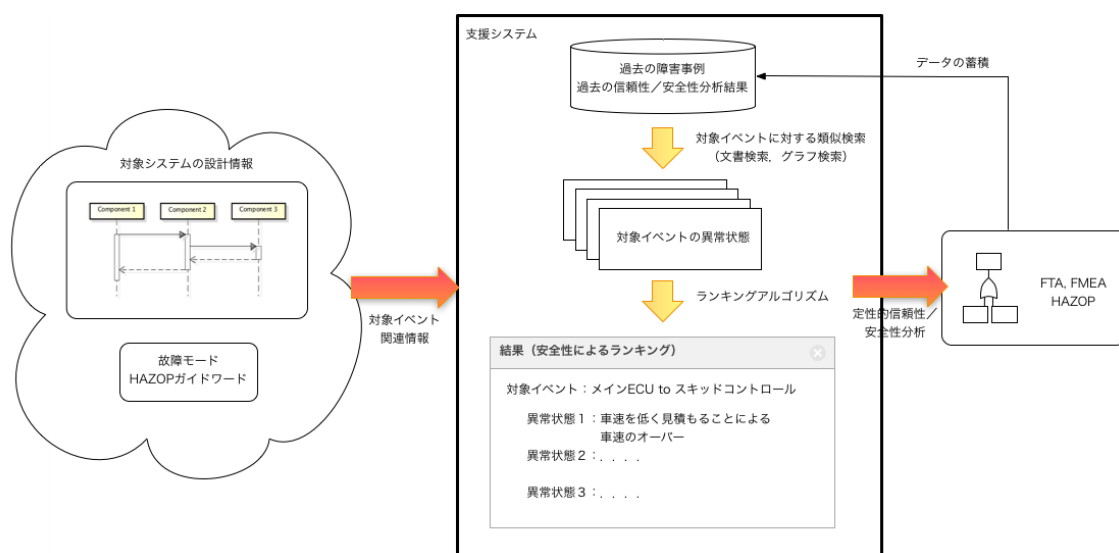


図1 支援システムの概要

## 2. 研究内容

本研究では、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、優先順位をつ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。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および図3は作成する支援ツールのユースケースを示している。図2は過去の分析結果を学習するフェーズであり、(i) 一般的な文章および定性的信頼性／安全性分析に関連した文章から単語（タグ）を抽出およびベクトル化、(ii) 入力された過去の設計書およ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，(i) 入力された設計書の分解およびタグ付け，(ii) タグに基づいた過去の分析例との類似検索，(iii) 類似度と重要度のスコアリングを行う．

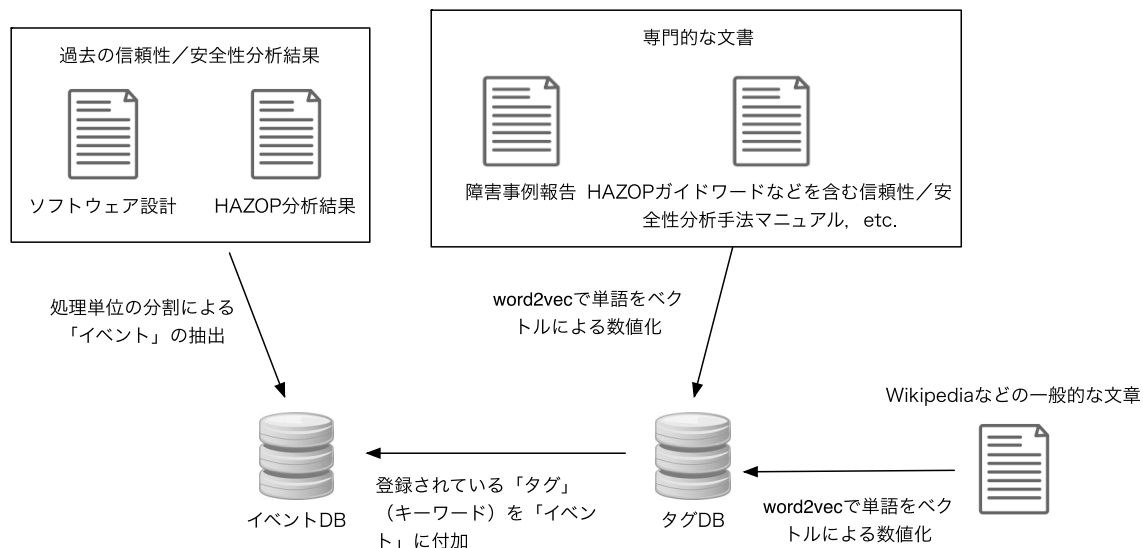


図 2 支援システムにおける学習フェーズ

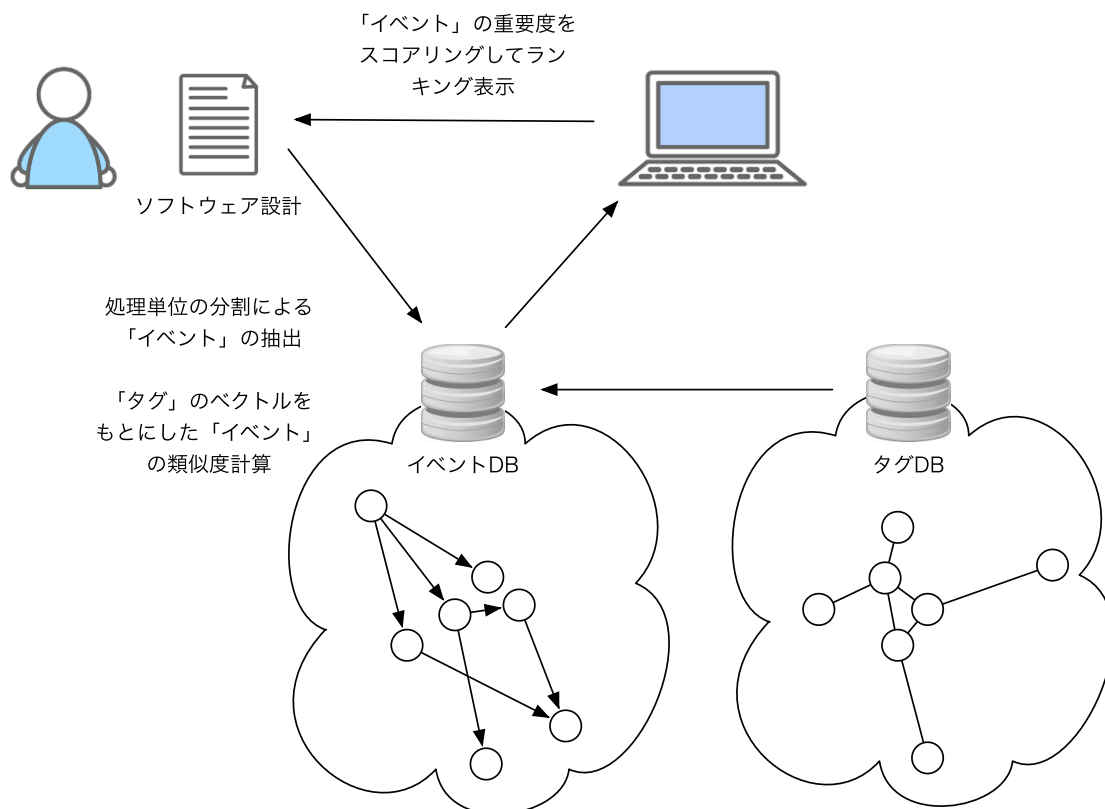


図 3 支援ツールにおける評価フェーズ

上述のシステムを開発するために、次の研究課題を設定する。

研究課題 1：検索手法の調査・開発

- ・構造化・非構造化データを統一的に扱うためのデータ構造の定義
- ・類似検索を実現するための類似度の定義

研究課題 2：スコアリング手法の調査・開発

- ・単語（タグ）の数値ベクトル化
- ・類似度に基づいた重要度スコアリング手法の検討

研究課題 3：分析例の蓄積

- ・学習データの収集
- ・信頼性／安全性分析結果（UML 設計書，HAZOP 分析結果）の収集

研究課題 4：ツールの開発・実装

研究課題 5：有効性検証

### 3. 研究成果

#### 研究課題 1「検索手法の調査・開発」

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統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言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既定することで、すべてのデータを統一的に扱う仕組みを構築した。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った。

##### ①データ構造の定義

テキスト文書および UML シーケンス図を格納するデータ構造を図 4 の UML クラス図で表現した。

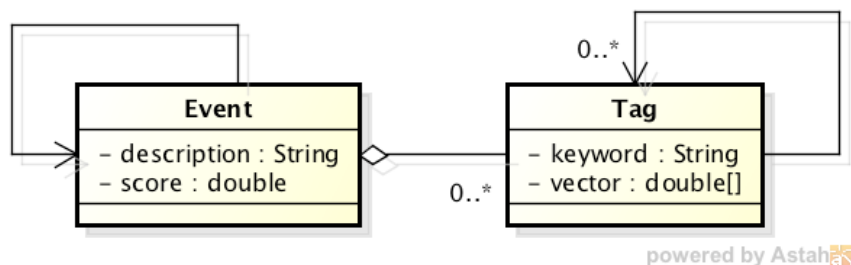


図 4 データ構造のクラス図

テキスト文書および UML シーケンス図は以下の単位の「文章」に分解され Event クラスを生成する。



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

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, 「スコアリング手法の調査・開発」で説明する) 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. さらに, 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で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), word2vec によって算出されたベクトル (vector) の属性を有する. description から自動的に日本語の単語を抽出するために MeCab による「わかち書き」を適用した.

## ②類似検索アルゴリズム

Event クラス間の類似度を算出するために, 本研究では word2vec によるタグの数値ベクトル化を行った (「研究課題 2」で説明する). 数値ベクトル化された Tag クラスをもとに, Event クラスの数値ベクトルは Tag クラスの TF-IDF 値をもとに以下の計算で得られる.

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

ここで,  $\text{tfidf}_{ij}$  は Tag クラス  $i$  の Event クラス  $j$  における TF-IDF 値である. 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タグに対して以下の式で定義されるコサイン類似度を用いた.

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

ここで,  $S_A^n, S_B^n$  は Event クラス A および B を中心に  $n$  ホップ先までのイベントを含めた集合を表す.

## 研究課題 2 「スコアリング手法の調査・開発」

研究課題 2 では, Tag クラスの特徴を数値的に表現する手法と, Event クラス間の類似度をもとにした重要度スコアの定義について検討を行った.

### ①Tag 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った. 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) と呼ばれる以下のアーキテクチャをも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．

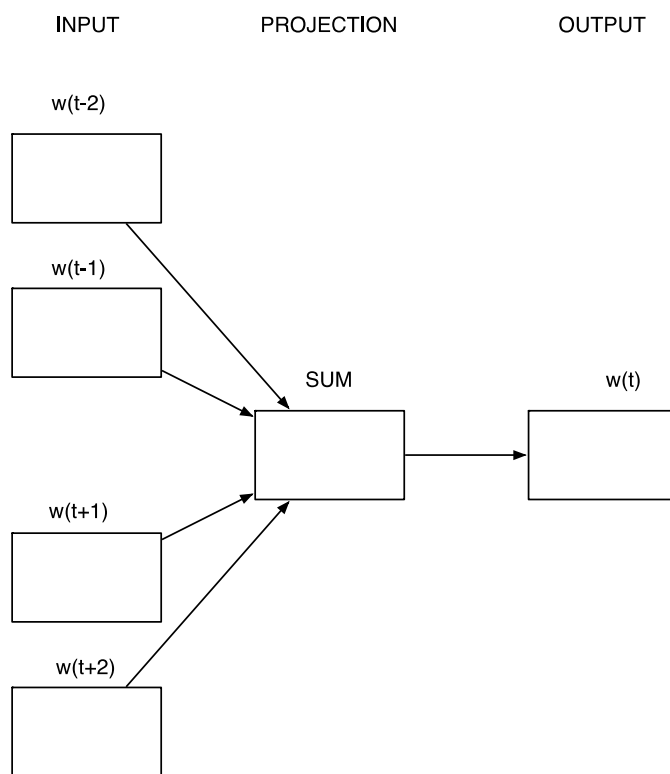


図5 CBOW のアーキテクチャ

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．日本語コーパスには Wikipedia における文章，IPA が出版する電子書籍，国際会議にて収集したデータをテキスト化して利用した．

## ②重要度スコアリング

重要度の算出については，文書（イベント）間の類似度と過去の HAZOP 分析結果に予め付加されている重要度（HAZOP スコア）から算出する．

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）

$I(\cdot)$ : 指標関数

$$I(A) = \begin{cases} 1 & A \text{ is true} \\ 0 & A \text{ is false} \end{cases}$$

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  $\theta$  以上の HAZOP スコアの平均

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

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も用いた. 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に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. いま、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.  $\Omega$  をノード全体の集合、ノード  $i$  のリンク先を  $\Omega_i$  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  $i$  を訪問している時に選択できる行動を

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

とする. ここで、 $\Omega_i$  はリンクをたどって移動するノード、 $\text{rnd}$  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. さらに、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  $\beta^n$  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できる. 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.

いま、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.

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

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

上記の最適性方程式を満足する  $V(\cdot)$  および  $W(\cdot)$  を見つけることで、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。

### 研究課題 3 「分析例の蓄積」

本研究では「良い分析例」を「過去の分析例」としてデータベースに蓄積し、教師データとして利用することで、より良い精度で分析を支援することが期待される。そのため、信頼性／安全性分析の既存手法および最新手法の包括的な調査を行う。

実際に、分析データとして JasPar (Japan Automotive Software Platform and Architecture) で実施したソフトウェア安全要求・設計仕様書を入手しこれを検証に利用した。また、word2vec の学習データとして以下の IPA が発行する電子書籍を入手し、word2vec の学習データ（日本語コーパス）として利用した。さらに、11 件の国際会議に参加し、そこで紹介されたすべての論文（213 本）をコーパスとして利用した。

### 研究課題 4 「ツールの開発・実装」

開発したツールは GUI によるフロントエンドとデータベースからなる。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」の機能を有する。

「タグ DB 登録」では word2vec からの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。実証実験においては、Wikipedia、IPA が発行する電子書籍、国際会議にて収集したデータのテキストを利用し、667644 単語に対するベクトル化を行っている。また、バイナリ形式にすることにより、667644 単語のベクトルデータを 2.68GB のファイルサイズで実現できた。

「イベント DB 登録」では XML 化されたドキュメントまたは EA (Enterprise Architecture) 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 Microsoft Excel で管理されることも多い。そこで、Excel から入力用の XML ファイルを生成するスクリプトの作成も行った。一方、シーケンス図は EA がエクスポートする XML ファイルを直接解析して UML コンポーネントおよびそれらのリンクを登録できる。

「類似度検索」では、まず検索をするもとのイベントを、登録してあるイベントから選択し、指定した検索範囲に該当するイベントすべてに対する類似度を算出する。ここでは、コサイン類似度 (COS similarity) およ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) の二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」(図 6) は開発したツールのメインの機能であり、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 によるものを適用できる。また、類似検索と同様に、類似度にはコサイン類似度および関連するイベントを考慮したコサイン類似度の二種類が利用できる。

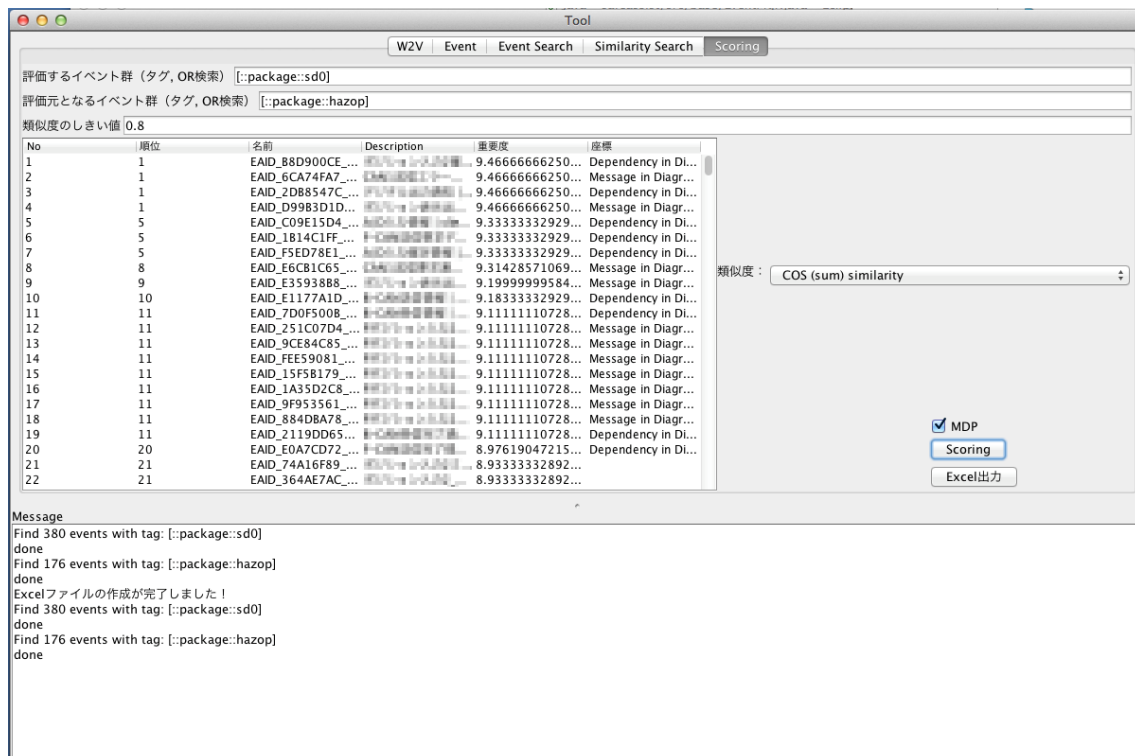


図6 重要度スコアリング

## 研究課題5「有効性検証」

実際の企業（A社，B社）における設計者および安全性分析者によって作成したツールの評価を行った。

実証分析（A社）

### 利用データ

利用する設計データは，自動車のシフト制御に関するシーケンス図であり，二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う．重要度スコアリングのもととなる 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）で実施されたもの（HAZOP1）と，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）で比較を行った．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．この分析結果には，安全性分析者の手で予め，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る．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 のレベルに応じた5段階の評価を行った。

## 分析結果と考察

HAZOP1 を使った評価では、シーケンス SD0, SD1 とともに類似度の閾値 0.8 として評価を行った。SD0, SD1 ととも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 程度のイベントを分析対象から外すことができているという結果が得られた。

一方、スコアリングのもととして 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ことなり、全体的な類似度が低下した。そのため、HAZOP1 と比較すると有効なスコアリングが得られないという結果となり、もとなる 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にいたる必要があることがわかった。

## 実証分析 (B 社)

### 利用データ

利用する設計データは、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) で比較を行った。シーケンス図をイベントに分解して登録した際のイベント数は 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。

## 分析結果と考察

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」およ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」などが重要であるという指摘があり、精度についての問題が指摘された。その要因として 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原因と考えられる。

## 4. まとめ

本研究では、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解析した。特に、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、それをを用いた重要度分析手法について検討を行った。通常の文書間における類似度分析とはことなり、ソフトウェア設計に使われる文章は、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。そこで、本研究では単語そのものをベクトル化する word2vec という技術に注目し、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、辞書の整理をすることでより精度が高まること

がわかった．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 分析結果に予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た．重要度をつけることによって，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実現できた．しかしながら，精度がまだ十分ではなく実用に向けては，辞書の整理などによる類似度を高めることで，重要度の評価精度を高める必要がある．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．さらには，同一ドメインに対する HAZOP 分析がない場合への対応なども考える必要がある．本研究では，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予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ような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，入力およ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．特に，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が視覚的にすることで，重要な部分の特定を助け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 になっており，これを生成するための労力が必要である点が指摘される．これらについては，統合開発環境との融合により解決すべき問題でもある．さらに，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な状況においては分散処理などの検討が必要である．

## 1 研究の目的・背景と期待される効果

### 1.1 研究目的とその背景

近年、CPS (Cyber Physical System) に代表されるように、実世界とコンピュータシステムの結びつきが非常に強くなってきている。CPS では、実世界 (Physical System) のセンサーネットワークなどで得た情報を、何らかのモデル化を伴ったサイバー空間 (Cyber System) 上でコンピュータ処理することにより、より高いレベルでのサービスを提供することを目指している。例えば、自動車における自動運転制御などは CPS における典型的な事例であり、実際に、数年後には自動車における自動運転の機能が一般ユーザへ提供されるまでに至っている。

このような CPS の発展により、情報処理システムに対する重要性がこれまで以上に必要視されるものと考えられる。自動車の自動運転を例に挙げれば、従前までは運転者の補助的な役割しか果たしていなかった情報システムが、運転者に代わって主体的な役割を担うことになる。言い換えれば、これまで運転者が責任を負っていた「安全」に関しても、情報システム自体がその責任を負う必要がある。

通常、CPS はセンサーとアクチュエータ、およびそれらを制御する組込みシステムから構成される。現在においても、組込みシステムは我々の生活の様々な場面で広く利用されている一方で、システム自体が必ずしも安全を保証できるような仕組みが実現できていない。例として、車載システムにおける制御関係の障害は一定のペースで発生し続けており、その原因のほとんどが組込みシステム内のソフトウェア制御部分における欠陥によって引き起こされていることはよく知られた事実である。

ソフトウェアにおける欠陥が発生する要因はいくつか存在するが、設計時における障害事象の「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とに限定される。一方で、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、ソフトウェアの開発期間内で完全に予防できることは極めて困難である。

考慮漏れによる欠陥の挿入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必要となる。しかし、一度でも経験したことがある事象であればある程度容易に想定できるかもしれないが、これまでに未経験である事象の発生を予め想定することは原理的に難しい。考慮漏れによる欠陥防止で最も難しいのはリスクを事前に想定することであり、ソフトウェアの安全性を確保するためには常に「どのようにすれば想定外を想定できるか」という大問題に直面している。具体的な手法として、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、人命に関わる事故など) 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り、元々化学プラントなどのプロセス工学における安全性分析に利用されていたものが、組込みシステムや航空宇宙システムの安全性分析でも頻繁に利用されるようになったものである。その考え方は故障モードを用いた FMEA 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表す自然言語表現を用いて、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値に依存するという問題である。FTA、FMEA、HAZOP は、本来、経験したことの無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いえる。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ところ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、「経験値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したり、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。

本研究では、このような問題点を部分的に解決も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新しい技術の提案を行い、開発現場において分析を支援するためのツール開発を行う（図 1-1 参照）。具体的に、過去の情報（設計や障害事例など）をデータベース上に蓄積し、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）から、関連する障害シナリオを抽出した上でそれら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。特に、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、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。先にも述べたように、定性的信頼性／安全性分析では想定外を事前に想定しなければならないため、分析者のスキルが分析結果に大きく影響する。一方で、経験豊富な分析者にとっても、この種の信頼性・安全性分析は限られた資質と資源で実施する必要があるため、過去の知見を効率よく利用できる支援ツールの開発は非常に重要であると考えられる。

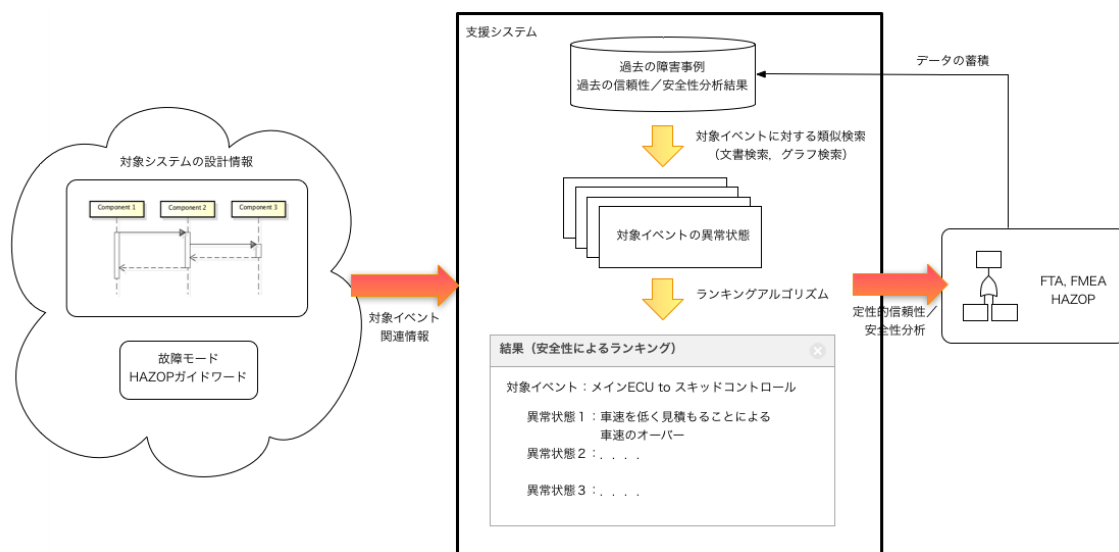


図 1-1 開発システムの概要

## 1.2 期待される効果

本研究で開発予定のツールを用いることにより、これまでエキスパートによって試行錯誤的に行われてきた信頼性・安全性のリスク分析を半自動的かつ系統的に実施することが利点として挙げられる。形式的方法において記述される故障モードに加え、データベース上に格納されたガイドワードや障害発生原因となるシナリオに関する情報の優先順位を考慮することで、FMEA や HAZOP をより効率的に実施することができる。特に、本研究では過去の経験を非構造型データベースとして蓄積し、データマイニング手法を応用することで定性的信頼性／安全性分析を支援するツールの試作を行う。現在、機能安全規格を設計に反映するために、開発現場ではリスク分析に関する大変な労力が必要とされている。本研究で開発予定のツールにより得られる効果として、(i) 経験の浅い分析者によるリスク分析における考慮漏れが防げること、(ii) リスク分析に関する工数の削減が期待できることが考えられる。リスクの考慮漏れを防止するためには、過去の類似したリスク分析事例をすべて抽出する網羅性が重要となる。一方で、網羅的に抽出されたリスク分析結果をすべて考慮することは、工数の増大を引き起こす原因となる。よって、リスク分析事例の優先度の計算を行い、重要な分析結果に対して優先順位をつけることで、リスク分析に関する工数を飛躍的に削減することが期待される。このようなデータマネジメント技法の精度は、扱うデータ自体の性質や検索アルゴリズムの性能に大きく依存するため、信頼性・安全性分析において適用されたデータを用いて開発ツールの有効性を検証することが必要となり、そのような検証を通じて実際の定性的信頼性・安全性実務への適用可能性が明確になるものと考えられる。定性的信頼性／安全性分析に対するこのような取り組みは、事例データベースの作成などでは部分的に試みられていたものの、データマイニング手法を応用した統計的分析と組み合わせることはこれまでに行われていない。その観点からも本研究は挑戦的な研究プロジェクトであり、本研究プロジェクトを成功させることにより、信頼性／安全性の定性分析に対する知能化を世界に先駆けて実現できるものと期待している。

## 2 実施内容

### 2.1 研究アプローチ

#### 2.1.1 研究の全体像

本研究では、対象システムの設計情報と過去の情報（過去に開発した類似システムの設計情報や障害事例）から対象システムの異常状態を予測し、優先順位をつけて分析者に提示する支援ツールを開発することを目標とする。特に、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、上位レベルでの定性的信頼性／安全性分析を支援するために寄与する。

具体的に、ここで開発する支援ツールは過去のリスク分析例（特に HAZOP 分析結果）に基づいて、現在の類似した設計に対するスコアリングを行う。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内容を評価（スコアリング）するフェーズに分けられる。図 2-1 および図 2-2 は作成する支援ツールの概要を示している。図 2-1 は過去の分析結果を学習するフェーズであり、(i) 自然言語によって記述された設計文章（ドキュメント）および定性的信頼性／安全性分析に関連した文章から単語（タグ）の抽出およびベクトル化、(ii) 入力された過去の設計書およびリスク分析例の分解と登録されている単語を特徴付け（タグ付け）するための学習、をそれぞれ行う。図 2-2 は現在開発予定の設計書を入力して評価を行うフェーズを表しており、(i) 入力された設計書の分解およびタグ付け、(ii) タグに基づいた過去のリスク分析例との類似性検索、(iii) 類似度と危険度スコアによる設計評価、をそれぞれ行う。

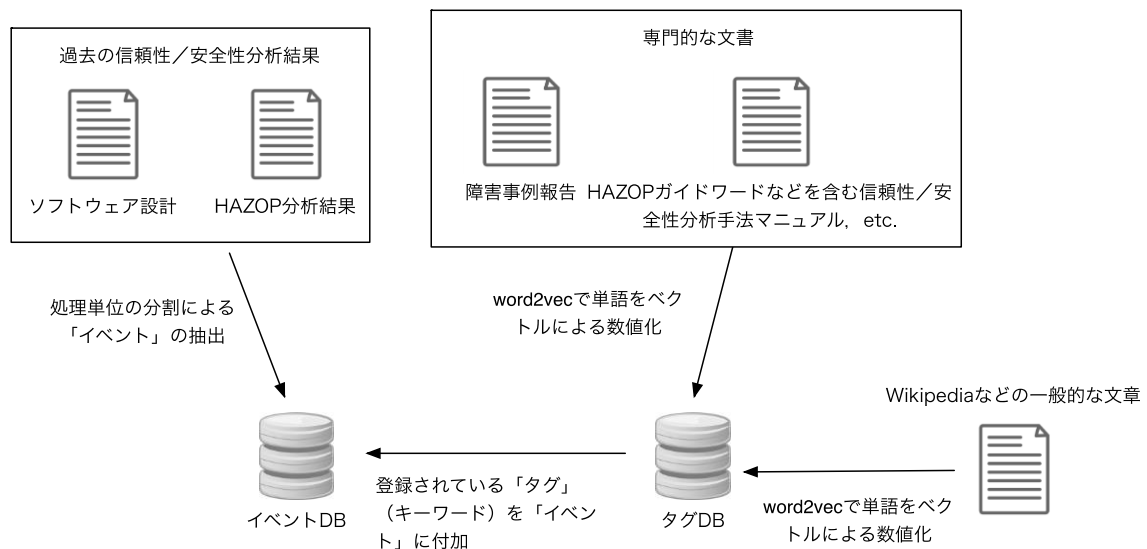


図 2-1 開発システムにおける学習フェーズ

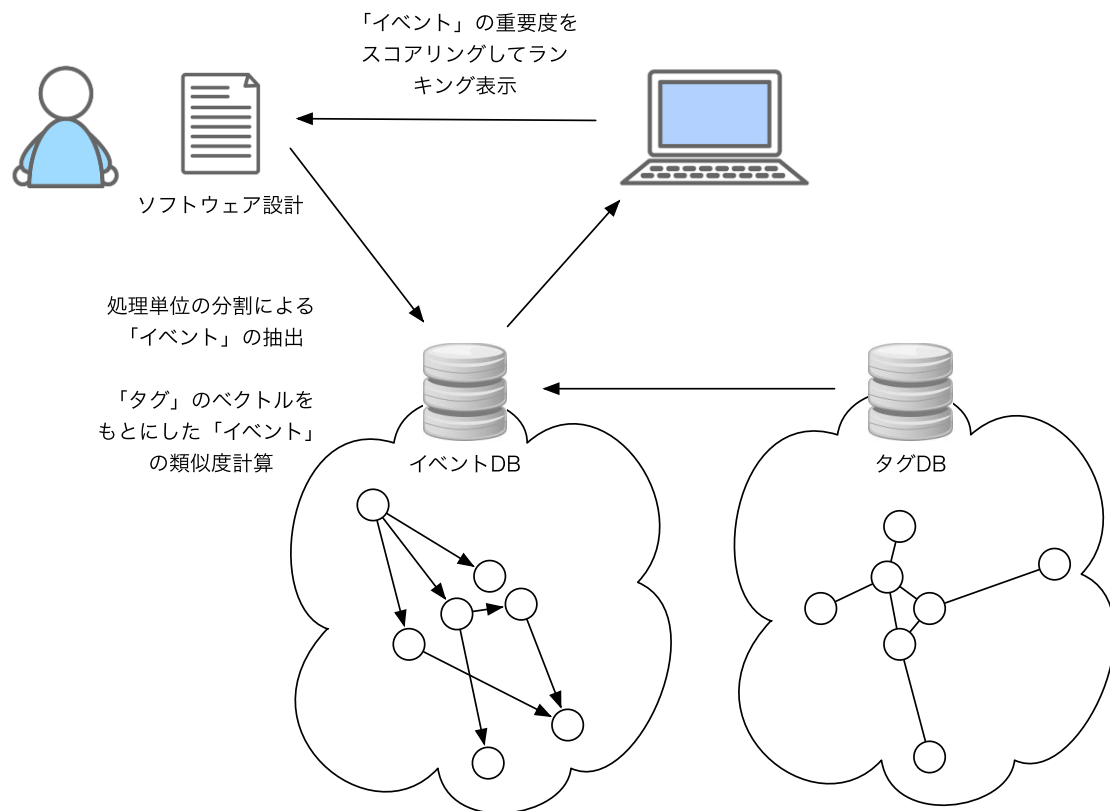


図 2-2 開発ツールにおける評価フェーズ

### 2.1.2 関連するこれまでの研究について

一般的に、リスク分析を支援するためには過去の分析情報（障害事例など）が蓄積されたデータベースを構築することが考えられる．実際に IPA/SEC でも「情報処理システム高信頼化教訓集」として、これまでに起こった障害事例をデータ化している．このような取組は、マクロな意味において障害分析の大きな支援にはなり得るが、実際の企業で行われている個別のシステム開発の現場において、実際に集計されたデータと分析するシステムの関連をマッチングする仕組みがなければ、データを直接的に活用しながら定性的信頼性／安全性分析を行うことは不可能である．このため、IPA/SEC では、「教訓」として重要な知見を事例集としてまとめたり、早見表のようなマトリクス形式の指標を作成したりすることで、容易に過去の事例を活用するための工夫を行っている．しかしながら、収集された知見の中に細かな情報が埋もれていたり、過去の事例と分析対象とするシステムの関連性を見つけれない場合、このように原始的な方法では信頼性・安全性に関する重要なポイントを見落とす可能性が大いに考えられる．データマイニング技術と旧来からあるデータベース技術の大きな違いは、データ整理の方法論にあると言っても過言ではない．従来のデータベースでは、蓄える情報を予め整理した上で格納する必要があった．IPA/SEC による「教訓」や早見表なども、ある意味でこの考え方に基づいている．これは検索の効率性の観点から有効であるが、予め情報を整理するためのコストや、整理の仕方によっては逆に検索が難しくなることもある．また、もともと情報の整理が難しい図やグラフなどの非構造的なデータに関し

では、有効に機能しないことが広く知られていた。このような問題に対して、データマイニング技術では、データを予め分類・整理することなくとりあえず蓄積しておき、必要に応じてアドホック的に情報を抽出するための分析を行う。最近では、統計的な学習モデルの発展と相まって、非構造的なデータからの情報抽出技術や、経験豊富な技術者の知見を学習・分析するための技術が続々と開発されている。

広島大学では2013年度ソフトウェア工学分野の先導的研究支援事業に関する委託契約において、テスト段階以降の下流工程におけるソフトウェア信頼性評価技法について研究を行っている。一方、開発上流工程の定量的信頼性評価では定性的分析を詳細に行うことに主眼が置かれており、定量的な信頼性評価を行うための方法論が未だに整備されていないという問題がある。この最も大きな要因として、ソフトウェアの潜在的な故障原因であるリスクを洗い出すための方法において、さえ汎用的かつ実務的なツールが未整備の状況であり、信頼性・安全性分析を実際の実験現場で実施する上で障害となっていることが挙げられる。よって、上流工程における定性的信頼性・安全性分析手法の開発は、下流工程における信頼性評価手法と補完的な位置づけにあると言える。

本研究実施者らの研究グループでは、これまでにソフトウェア信頼性評価の分野において、ソフトウェアメトリクスと定量的なソフトウェア信頼度の関連について分析を行ってきた。例えば、ソフトウェアソースコードやPDG (Program Dependency Graph) など、ソフトウェアそのものに内在する情報から、テキストやグラフなどの非構造化データを抽出し、データマイニング手法を適用することでソフトウェアの信頼度を定量的に評価する技術を開発している(2013年度ソフトウェア工学分野の先導的研究支援事業に関する委託契約)。また、信頼性工学以外の領域においても、マルコフ決定過程や強化学習など、データマイニングによく利用される技術の応用的研究を継続的に実施してきた。具体的には、各種アプリケーションを省電力モードに移行する時期を決定するための適応的アルゴリズムの開発や、Web ページランキングアルゴリズムの提案などを行ってきた。一方で、定性的な信頼性/安全性分析については、2009年から継続的に、FTA、FMEA 分析に基づいたソフトウェア設計情報を活用した信頼性評価技術を研究しており、従来までに得られた研究成果を本研究における定性的信頼性/安全性分析に直接応用することができる。

## 2.1.3 研究目標と研究課題

### (1) 研究目標

本研究では、設計の上位レベルにおける定性的信頼性/安全性分析支援技術の開発を行う。特に、先に述べた産業界における二つの問題点を鑑みて、実務経験の少ない分析者でもリスクの考慮漏れが起きないように、網羅的な信頼性・安全性分析を支援する仕組みを提供する。さらに、分析に対するコストを効果的に削減するために、安全性あるいは信頼性の観点からリスク発生事象(システム異常状態)の優先順位を自動的に与えるようなスコアリングの仕組みも考案し、これらの学習アルゴリズムをもとにした分析支援ツールの開発を行う。

具体的には、近年、発達が目まぐるしいデータマイニング技術の適用を考える。データマイニング技術は、蓄積されたデータから価値のある情報を抽出する手法である。古くは統計によるデータ解析やデータベース技術として知られていたが、近年になって、機械学習技術、テキスト処理などの非構造的なデータを扱う技術や、スケーラブルなデータ処理技術と融

合することにより、多様かつ膨大なデータから情報を抽出することが可能となってきた。特に最近、この分野は「ビッグデータ」のキーワードで注目をあつめており、数多くの分野で応用されている全世界的に進展著しい技術である。

IEC 61508 や ISO 26262 に代表される機能安全規格がソフトウェア開発にも浸透しており、設計段階における安全性分析の必要性が高まっている。同時に信頼性／安全性分析に要する工数が極端に増加しており、開発期間のかなり大きい割合を示すようになってきている。先にも述べたように、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を対象となるすべての故障モードを適用したり、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といったマニュアル分析であるため、ソフトウェアの規模や複雑性の増加に伴いその組み合わせを網羅するために多くの工数が必要とされる。ソフトウェアの定性的な信頼性／安全性分析を行うことは必要不可欠である一方で、現状においてはその効率化さえもままならない状況にある。よって、システムの設計情報（UML / SysML）からシステムティックに関連する障害シナリオを抽出した上で、それらを重要度に従ってランキングすることは、試行錯誤的に実施されている信頼性／安全性のマニュアル分析の見通しを良くするだけに留まらず、潜在リスクに対する定性分析の大幅な工数削減に繋がるものと考えられる。

本研究では、過去の経験を非構造型データベースとして蓄積し、データマイニング手法を応用することで定性的信頼性／安全性分析を支援するツールの作成を行う。定性的信頼性／安全性分析に対するこのような取り組みは、事例データベースの作成などでは部分的に試みられていたものの、データマイニング手法を応用した統計的分析と組み合わせることはこれまでに行われていない。その観点からも本研究は挑戦的な研究プロジェクトであり、本プロジェクトを成功させることにより、信頼性／安全性の定性分析に対する知能化を世界に先駆けて実現できるものと考えている。データマイニング手法と定性的信頼性／安全性分析の融合は、2012 年以降、PHM など特定分野の国際会議で散見される程度であり、本研究を成功裏に終えることで、今後も大きな発展が望める分野でもある。

## (2) 研究目標に向けた研究課題の設定

本研究では、対象システムの設計情報と過去の情報（設計情報や障害事例）から対象システムの異常状態を予測し、優先順位をつけて分析者に提示する支援ツールの開発を目標とする。特に、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、上位レベルでの定性的信頼性／安全性分析を支援する。

具体的には次の研究課題を考える。

- 検索手法の調査・開発
  - 構造化・非構造化データを統一的に取り扱うためのデータ構造の定義
  - 類似検索を実現するための類似度の定義
- スコアリング手法の調査・開発
  - 単語（タグ）の数値ベクトル化
  - 類似度に基づいたスコアリング手法の検討
- 分析例の蓄積
  - 学習データの収集

- 信頼性／安全性分析結果（UML 設計書，HAZOP 分析結果）の収集
- ツールの開発・実装
- 有効性検証

## 2.2 研究の活動実績・経緯

進捗予定管理表の最終実績を以下に示す。

進捗予定管理表

研究機関名	広島大学	研究責任者名	土肥 正										
研究テーマ名	2015年度ソフトウェア工学分野の先導的研究支援事業 研究テーマ名												
作業項目		6月	7月	8月	9月	10月	11月	12月	1月	2月	進捗状況	備考	
1. 研究課題	予 実												
(1) 検索手法の調査開発	予 実												
検索手法の調査開発	予 実										100%		
検索手法の実装	予 実										100%		
検索手法の検証	予 実										100%		
(2) スコアリング手法の調査・開発	予 実												
判別分析の調査・開発	予 実										100%		
判別分析の実装	予 実										100%		
スコアリング手法の調査・開発	予 実										100%		
スコアリング手法の実装	予 実										100%		
スコアリング手法の検証	予 実										100%		
(3) 分析データの蓄積	予 実												
国際会議への参加と情報収集	予 実										100%		
調査した手法による分析と結果の蓄積	予 実										100%		
(4) ツールの開発・実装	予 実												
システム設計	予 実										100%		
RFPの作成	予 実										100%		
検索手法の再実装	予 実										100%		
スコアリング手法の再実装	予 実										100%		
(5) 有効性検証	予 実												
協力企業の選定	予 実										100%		
分析結果のデータベース化	予 実										100%		
ツールを用いた分析	予 実										100%		
従来手法との比較	予 実										100%		
2. 外注	予 実												
(1) 仕様作成・確認	予 実										100%		
(2) IPAによる仕様確認	予 実										100%		
(3) 公告・入札・契約	予 実										100%		
(4) 検収	予 実										100%		
(5) 外注実施	予 実												
詳細設計・実装	予 実										100%		
テスト	予 実										100%		
3. 中間報告	予 実												
(1) 準備	予 実										100%		
(2) 中間報告	予 実										100%		
4. 最終成果のとりまとめ	予 実												
(1) 成果報告書の構成案	予 実										100%		
(2) 最終報告準備	予 実										100%		
(3) 最終報告	予 実										100%		
(4) 成果報告書の作成	予 実										100%		
5. 成果物の提出	予 実										100%		



研究の進捗状況は以下の方針によって管理した。

#### 研究の進捗管理

1. 研究責任者は、研究チーム全員が参加する進捗報告会を随時開催し、研究の進捗状況を確認および研究課題に対する成果の確認し、研究内容の助言・指導を行った。また、出張や学会参加報告も進捗報告会にて行った。進捗報告会を開催する場合には議事録を残すことを心掛けた。
2. 研究責任者は、指定月初旬に IPA 担当者に研究成果の進捗報告を行った。ただし、重大な問題発生時や計画変更時に即時に報告することを心掛けた。

#### 外注管理

1. 研究責任者は外注管理に関して以下の管理を行った。
2. 定性的信頼性/安全性解析支援ツールの委託開発部分についての仕様書の内容確認を行い、IPA へ仕様確認手続きを依頼した。
3. 外注先に対して隔週に進捗状況を報告させ、逐次内容を確認した。

以下に二ヶ月毎の進捗報告を記載する。

#### 報告期間 2015 年 6 月～2015 年 7 月

##### ①スコアリング手法の調査・開発

スコアリング（およびクラスタリング）に利用可能な手法の調査を行った。特に、テキストマイニングなどで、文書のクラスタリングに利用される「トピックモデル」についての学術的成果に関する調査を中心に行った。さらに、特徴量の抽出に関する手法として、「カーネル法による PCA」および「ディープラーニング」に関する調査を行った。

##### ②リスク分析例の蓄積

国際会議へ参加し、本プロジェクトで利用可能な最新の信頼性／安全性分析手法の調査および信頼性／安全性分析事例の調査を行った。7 月末までに参加した具体的な国際会議は、DSN 2015（岡村）、Ada-Europe 2015（土肥）、COMPSAC 2015（土肥）であり、IoT におけるサイバーセキュリティに関する知見や、LLC（Long Latency Crash）を用いた分析事例なども事例データベースに取り入れる必要があることがわかった。

##### ③ツールの開発・実装

データベース部分と GUI 部分など、外注として業者に発注する部分に対する仕様を決定した。また、広島大学の入札プロセスに先立って IPA-SEC による仕様確認が行われた。

##### ④有効性検証

本プロジェクトで作成するツールを用いた信頼性／安全性分析に関して協力してもらえ

る企業の調査を行った。7月末時点では組込みシステムを開発している二社から前向きな回答が得られた。

#### ⑤進捗報告会

6月5日、6月11日、7月7日、7月27日にそれぞれ進捗報告会を実施した。

#### ⑦学会参加

国際会議 DSN2015, Ada-Europe2015, COMPSAC2015 に参加した。

### 報告期間 2015 年 8 月～2015 年 10 月

#### ①検索手法の調査開発

検索手法に利用可能な方法論の調査を行った。日本語および英語のテキストに対する処理が中心となるため、日本語の分かち書きに利用可能なツールの調査を行った。特に、広く利用されている MeCab を Java へ移植した Sen, lgo などが利用できることがわかった。また、それらに実際の障害事例などが記述された文章を入力し、適切なワードやイベントの抽出するためのテストを行い、アルゴリズムを試作した。

#### ②スコアリング手法の調査・開発

文書クラスタリングとして LDA (Latent Dirichlet Allocation) トピックモデルを試作し、テストデータによる分類を行った。さらに、スコアリング手法として、マルコフ連鎖により PageRank および、報酬構造を考慮したマルコフ決定過程によるランキング手法の調査と試作を行った。

#### ③分析例の蓄積

国際会議へ参加し、本プロジェクトで利用可能な最新の信頼性／安全性分析手法の調査および信頼性／安全性分析事例の調査を行った。8月～10月末までに参加した具体的な国際会議は、QRS 2015 (土肥), EDCC 2015 (土肥), ICSME 2015 (土肥) であり、車載システム設計の安全性分析において考慮すべき設計情報 (入力データ) を検討するための情報を得た。また、IPA の「情報処理システム高信頼化教訓集 (製品・制御システム編)」の一部のデータからトピック分類のためのテストデータを作成した。

#### ④有効性検証

本プロジェクトで作成するツールを用いた信頼性／安全性分析に関して協力してもらえ  
る企業二社を決定し、秘密保持契約のための事務手続きを行った。

#### ⑤進捗報告会

8月5日、9月18日に進捗報告会を実施した。

#### ⑥学会参加

国際会議 QRS 2015, EDCC 2015, ICSME 2015 に参加した。

## 報告期間 2015 年 11 月～2015 年 12 月

### ①検索手法の調査開発

検索手法の実装および検証を行った。9 月までの成果で MeCab による日本語形態素解析を利用することを決めていたが、Java での利用方法が複数種類存在するため、実装を行った上で検証を行った。MeCab そのものは Windows, MacOS, Linux のいずれのプラットフォームでも動作するが、Windows では、Java と連携させるための動的ライブラリのコンパイルなどが利用者に対する敷居が高いため、コマンドラインを介したインタフェースを最終的に実装した。また、MeCab の分かち書きの精度が検索精度に影響するため、MeCab に対する辞書を充実させることが必要となった。この点については、インターネットを通じて収集できる情報（Wikipedia の見出し語など）で MeCab 辞書を作成した。

### ②スコアリング手法の調査・開発

試作に基づいた検証を行った。その結果として、LDA トピックモデルを利用する場合、どの単語に注目するかがスコアリングに大きく影響することがわかった。また、同じ意味であるが使われ方が異なる単語（例えば、障害やエラーなど）をクラスタリングした上で分類することも必要になることがわかった。そこで、単語を直接ベクトルによって表現する手法を採用することとした。具体的には Word2Vec と呼ばれる手法を採用することにした。Word2Vec では単語をベクトル化する際に、自然言語で記述された文書が必要となる。これについては「分析例の蓄積」によって収集したドキュメント（論文原稿）や IPA/SEC が配布している電子 Book などを利用することとした。また、ベクトル化された単語をもとにした文書間の類似度計算として、単語和によるコサイン類似度と正準相関分析によるものを実装し、さらにそれらを報酬とするマルコフ決定過程によるスコアリング手法も実装した。

### ③分析例の蓄積

国際会議へ参加し、本プロジェクトで利用可能な最新の信頼性／安全性分析手法の調査および信頼性／安全性分析事例の調査を行った。10 月～11 月末までに参加した具体的な国際会議は、ESEM 2015（羅）、ISSRE 2015（岡村）、ASE 2015（岡村）、PRDC 2015（岡村）であり、安全性や信頼性分析に関連した論文の収集を行った。「スコアリング手法の調査・開発」で記述したように、これらのドキュメントは Word2Vec の学習データとして利用する。

### ④ツールの開発・実装

「検索手法の調査開発」「スコアリング手法の調査・開発」で実装したものをもとにして、実際のツール化のためのリファクタリングを行った。文書をある程度意味のまとまった単位で「Event」クラスに分解し、さらにその文書から MeCab を通じて単語を抽出し、それを「Tag」クラスとして全体の設計を行った。

### ⑤有効性検証

信頼性／安全性分析に関して協力してもらえらる企業二社から 11 月初旬にシステムの設計および安全性分析結果に関するデータを入手した。それらのデータ構造をもとにして上記の Event および Tag クラスの設計を行った。

#### ⑥進捗報告会

10 月 2 日に進捗報告会を実施した。

#### ⑦学会参加

国際会議 ESEM 2015, ISSRE 2015, ASE 2015, PRDC 2015 に参加した。

### 報告期間 2015 年 12 月～2016 年 1 月

#### ①分析例の蓄積

国際会議へ参加し、本プロジェクトで利用可能な最新の信頼性／安全性分析手法の調査および信頼性／安全性分析事例の調査を行った。12 月～1 月末までに参加した具体的な国際会議は, ASPEC 2015 (土肥)であり, 安全性や信頼性分析に関連した論文の収集を行った。これらのドキュメントは Word2Vec の学習データとして利用する。さらに, 有効性検証に向けて, IPA が発行している電子書籍から Word2Vec の学習データを取得した。また, JasPar (Japan Automotive Software Platform and Architecture) の安全性分析例を入手した。

#### ②ツールの開発・実装

11 月に引き続き「検索手法の調査開発」「スコアリング手法の調査・開発」で実装したものに基づいて, 実際のツール化のためのリファクタリングを行った。

#### ③有効性検証

協力企業二社からのデータを XML 化し, データベースへの登録作業を行った。また, システムテスト時にそれらのデータを用いた分析を行った。加えて, 同一ドメインの HAZOP 分析との比較を行うため JasPar におけるデータの一部もデータベースへ登録した。

#### ④学会参加

国際会議 ASPEC 2015 に参加した。

## 2.3 研究実施体制

開発ツールのうち, スコアリングアルゴリズム等の検索エンジンは広島大学で開発し, GUI インタフェースおよびデータベースは外部業者に開発を委託した。実例データに基づいた開発ツールの評価・比較は広島大学で実施した (図 2-3 参照)。

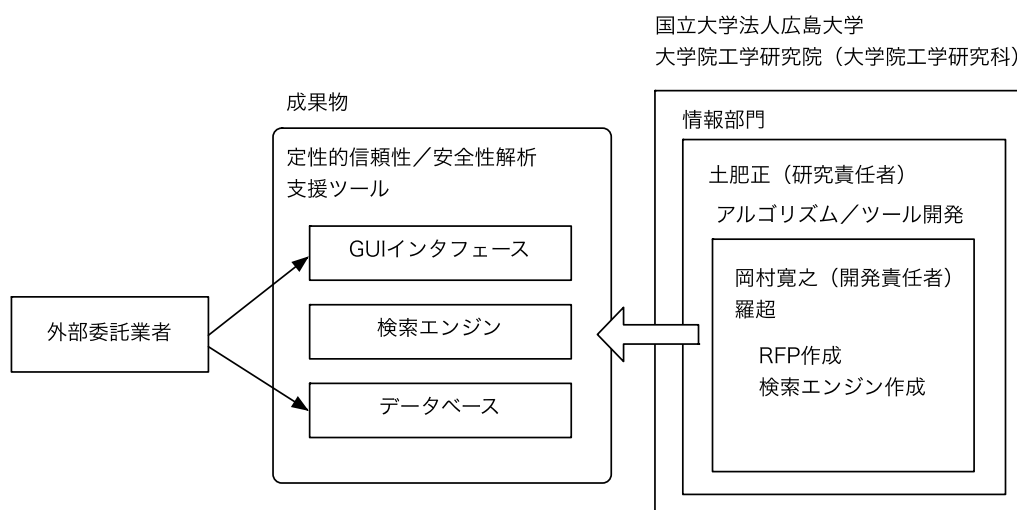


図 2-3 研究実施体制

表 2-1 研究実施体制(役割分担表)

実施機関名	学校法人 広島大学大学院工学研究院情報部門		
研究責任者	教授 土肥 正		
連絡担当者	同上		
主な実施場所 および研究者	広島大学東広島キャンパス 〒739－8527 住所 東広島市鏡山 1 丁目 4 番 1 号 連絡先 TEL 082-424-7698 FAX 082-422-7025 e-mail dohi@rel.hiroshima-u.ac.jp		
	氏名	職種	担当内容
	岡村 寛之	准教授	アルゴリズム／ ツール開発
	羅 超	研究員	アルゴリズム／ ツール開発

## 研究責任者のプロフィール

(ふりがな) 氏 名	どひ ただし 土肥 正
生年月日	1965 年 6 月 18 日
所属機関	国立大学法人 広島大学
所属 (部署名)	大学院 工学研究院 情報部門 (ディペンダブルシステム論研究室)
役 職	教授
<b>【学歴（大学卒業以降）】</b> 1989: 広島大学工学部第二類 卒業 1991: 広島大学大学院工学研究科博士課程前期 修了 1992: 広島大学大学院工学研究科博士課程後期 中途退学	<b>【職歴】</b> 1992: 広島大学工学部 助手 1992: UBC (カナダ) 客員研究員 1996: 広島大学工学部 助教授 2000: デューク大 (米国) 客員研究員 2002: 広島大学大学院工学研究科 教授
<b>【研究実績】</b> ソフトウェア信頼性工学，ディペンダブルコンピューティングの研究に従事．学術雑誌掲載論文 210 編，査読付国際会議論文 270 編，著書 48 冊（編著書 20 冊，分担執筆 28 冊）．ISSRE 2011（広島），ATC 2012（福岡），QR2MSE 2014（大連）等，内外で開催された 10 以上の国際会議における実行委員長を務める．PRDC 2012 プログラム委員長，COMPSAC 2013 トラックチェア，IEEE 主催国際会議 ISSRE，DSN，PRDC，SRE や ACM 主催国際会議 RACS，SAC 等のプログラム委員を歴任． <b>【主な論文・著書】</b> (1) “Enhancing performance of random testing through Markov chain Monte Carlo methods,” <i>IEEE Transactions on Computers</i> , vol. 62, no. 1, pp. 186–192, 2013 (with B. Zhou, and H. Okamura). (2) “Wavelet shrinkage estimation for NHPP-based software reliability models,” <i>IEEE Transactions on Reliability</i> , vol. 62, no. 1, pp. 211–225, 2013 (with X. Xiao). (3) “Estimating software intensity function based on translation-invariant Poisson smoothing approach,” <i>IEEE Transactions on Reliability</i> , vol. 62, no. 4, pp. 930–945, 2013 (with X. Xiao). (4) “SRATS: Software reliability assessment tool on spreadsheet,” <i>Proceedings of The 24th International Symposium on Software Reliability Engineering (ISSRE 2013)</i> , pp. 100–117, 2013 (with H. Okamura). (5) “Towards quantitative software reliability assessment in incremental development processes,” <i>Proceedings of 33rd International Conference on Software Engineering (ICSE-2011)</i> , pp. 41–50, 2011 (with T. Fujii, and T. Fujiwara,).	

(ふりがな) 氏 名	おかむら ひろゆき 岡村 寛之
生年月日	1973 年 2 月 21 日
所属機関	国立大学法人 広島大学
所属 (部署名)	大学院 工学研究院 情報部門 (ディペンダブルシステム論研究室)
役 職	准教授
【学歴（大学卒業以降）】 1995： 広島大学工学部第二類 卒業 1997： 広島大学大学院工学研究科博士課程前期 修了	【職歴】 1997： （株）CSK 入社 1998： （株）CSK 退社 1998： 広島大学工学 助手 2003： 広島大学大学院工学研究科 助教授 2006： デューク大（米国）客員研究員 2007： 広島大学大学院工学研究科 准教授

(ふりがな) 氏 名	ら ちょう 羅 超
生年月日	1984 年 9 月 6 日
所属機関	国立大学法人 広島大学
所属 (部署名)	大学院 工学研究院 情報部門 (ディペンダブルシステム論研究室)
役 職	研究員
【学歴（大学卒業以降）】 2007： 華中科技大学（中国）卒業 2012： 広島大学大学院工学研究科博士課程 前期 修了 2015： 広島大学大学院工学研究科博士課程 後期 修了	【職歴】 2008： （株）イクイップル・ソリューションズ（東京） 入社 2009： （株）イクイップル・ソリューションズ（東京） 退社 2015： 広島大学大学院工学研究科 研究員

### 3 研究成果

#### 3.1 研究課題 1「検索手法の調査・開発」

##### 3.1.1 当初の想定

###### (1) 研究内容

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統一的に扱う必要がある。つまり、構造化されたデータと非構造的データを同時に扱うデータ構造の定義が必要になる。そのため、データベースに格納するデータ形式について注意深く検討する必要がある。また、FTA, FMEA, HAZOP の手法を支援するためには、キーワードだけに基づいた検索では不十分である。例えば、HAZOP のガイドワードは more, less などの形容詞で表され、これらのキーワードから「もし、この数値が規定の範囲より大きい（小さい）場合は…」というシナリオを想起した上で分析を行わなければならない。そのため、more, less などの文字上のキーワードだけを用いた検索では、過去に同様な事例があったとしても該当するものを検索することは困難である。すなわち、異なるドキュメント間での類似検索が必須となる。文書類似検索は様々な手法がこれまでに提案されているが、用いる手法によってデータ形式が全く異なるため、類似検索手法についても併せて注意深く検討を行う必要がある。

###### (2) 想定問題と対応策

設計情報（UML/SysML）は非構造的データ（グラフ）で表されているため、そのようなデータを扱うための技術が必然的に必要となる。非構造的データ扱う手法としてカーネル法があり、これを用いることで類似したグラフの検索を行う技術を詳細に検討することが可能である。ツール上で分析したデータに対しても、ツール上で蓄積され、さらに再利用される必要がある。そこで、付加的なデータ（タグ）の付け方に関する方法論についても検討する。

##### 3.1.2 研究プロセスと成果

###### (1) 研究プロセス

研究は以下の作業項目からなる。

- ・ 検索手法の調査・開発
- ・ 検索手法の実装
- ・ 検索手法の検証

検索手法の調査・開発では、既存の類似検索手法についての調査を行い、その検索に必要なデータ構造と利点・欠点などの調査を行う。検索手法の実装および検索手法の検証では、類似検索を実装し、擬似的に作成したデータ（HAZOP 安全性分析結果と UML 設計図）を用いて本研究課題に適するかどうかの検証を行う。

###### (2) 具体的な研究成果の内容

###### ① 検索手法の調査・開発

検索手法の調査・開発では、まず、構造データと非構造的データを同時に扱うために、一



一般的なグラフを扱うデータ構造を参考にした定義を行った。具体的には、すべての文書を「イベント」と呼ばれるあるまとまった単位に分割し、関連するイベント同士を「リンク」で接続した。以下の表 3-1-1 にイベントの例を示す。

表 3-1-1 イベントの例

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

関連するイベント同士を接続するリンクについては、テキスト文書などでは前後の文章に基づいて、UML シーケンス図においては構成要素間の関連が定義されているのでその関連に基づいて、それぞれリンクを定義した。また、HAZOP 分析は機能別の階層的な構造となっているため、中項目などで同じグループとして定義されているもの同士にもリンクを定義した。

さらに、類似検索を行うためには、各イベントで記述されている文章の特徴を表す必要がある。一般的に、テキスト解析では現れる単語によって文書の特徴を表すことが行われる。例えば、「今日の天気は晴れ」という文章からは、「今日」、「天気」、「晴れ」という単語がこの文章を特徴付けるものとして抽出できる。ここでは、この単語を「タグ」と呼び、タグに基づいた類似検索を検討した。類似検索における二つの文章の類似度には、Jaccard 類似度（係数）[1]、コサイン類似度 [1]、LDA（潜在トピック分析）[2] などが用いられる。

Jaccard 類似度は共通するタグの含有率によって定義される。いま、文書 A におけるタグの集合を  $D_A$ 、文書 B におけるタグの集合を  $D_B$  とする。また  $|D_A|$ 、 $|D_B|$  をそれぞれ集合  $D_A$ 、 $D_B$  における要素数と定義すると、Jaccard 類似度（係数）は

$$\text{sim}(D_A, D_B) = \frac{|D_A \cap D_B|}{|D_A \cup D_B|}$$

で定義される。共通する単語  $|D_A \cap D_B|$  が多いほど類似度が 1 に近づく。

コサイン類似度は二つの文章が何らかの意味でベクトル化されている場合に適用できる。いま、文書 A と文書 B の特徴を表す実数値ベクトルが  $x_A$ 、 $x_B$  で与えられるとき、コサイン類似度はこれらのなす角  $\theta$  の大きさと与えられる。具体的には以下の計算を行う。

$$\text{sim}(x_A, x_B) = \frac{x_A \cdot x_B}{\|x_A\| \|x_B\|}$$

ここで  $x_A \cdot x_B$  は内積、 $\|x_A\|$  はユークリッドノルムを表す。つまり、なす角  $\theta$  のコサイン値（-1 から 1 の範囲で値をとる）を表す。文書の数値化（ベクトル化）には、TF-IDF（term frequency - inverse document frequency）[3] が導入されることが多い。TF-IDF は単語の出現頻度を加味した指標である。対象とする単語  $t_i$ 、 $i \in [1, k]$  と対象とする文章  $d_j$ 、 $j \in [1, D]$  があったとき、単語  $t_i$  の文書  $d_j$  における TF-IDF は

$$\text{tfidf}_{ij} = \text{tf} \times \text{idf}, \quad \text{tf} = \frac{n_{ij}}{\sum_k n_{ik}}, \quad \text{idf} = \log \frac{D}{\text{単語 } t_i \text{ を含む文書数}}$$

として算出される．ここで，TF に使われる  $n_{ij}$  は文書  $d_j$  における単語  $t_i$  の出現回数を表す．IDF は一般的な単語（すべての文章に共通して表れる単語）の重みを軽くする効果を持つ．対象とする単語を予め定義すると，それぞれの文書に対して TF-IDF 値によるベクトルが算出され，これらのコサイン類似度により文書間の類似度を得ることができる．

LDA は単語が出現する確率が「潜在的なトピック」によって決定するという仮定のもとで構築された確率モデルから，文書がどのトピックに属するかを統計的な推定によって数値化する．モデルの詳細については「研究課題 2 スコアリング手法の調査・開発」で言及する．

## ② 検索手法の実装および検索手法の検証

検索手法の実装および検索手法の検証では，上記の既存の類似検索を実装し検証を行った．Jaccard 類似度や TF-IDF によるコサイン類似度は比較的簡単な計算であるため，多くの場合で利用可能であるが，本研究ではイベントにおける文書が非常に短いため（例えば UML シーケンスにおけるメッセージなどはメッセージ名のみとなる），Jaccard 類似度や TF-IDF の利用が難しいことがわかった．同様に，LDA による解析もイベントにおける文書が非常に短いため，うまく機能しなかった．そこで，本研究ではこれらのアプローチとは異なる手法により，イベント間の類似度を定義する必要が生じた．具体的には，後述する word2vec [4] と呼ばれる単語（タグ）を直接数値ベクトル化する手法を用いることとし，それを利用できるデータ構造の定義を行った．また，数値ベクトル化されたタグを用いて文書の数値ベクトル化を以下のように行うこととした．

いま，文書  $d_j$  が単語  $t_i$ ,  $i \in [1, k]$  を含んでおり，各単語が数値ベクトル  $w_i$  で特徴付けられているとき，文書  $d_j$  の特徴を表す数値ベクトル  $v_j$  を以下のようにして算出する．

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

つまり，出現単語に対するベクトルの TF-IDF 値による加重和で表現する．このようにして算出される文書に対するベクトルから各文書間の類似度は以下のコサイン類似度で定義される．

$$\text{sim}_{ij} = \frac{v_i \cdot v_j}{\|v_i\| \|v_j\|}.$$

さらに，UML シーケンス図などの非構造的なデータに対する特徴を考慮するために，グラフカーネルのアイデアを取り入れた．グラフカーネルとはグラフ間の内積を定義する手法であり，ある種の類似度として利用できる．参考文献 [5] では，あるグラフ上のすべての部分グラフを用いたたたみ込みカーネルについて議論している．本研究では，部分グラフの考え方を応用して，あるイベントと別のイベントの類似度として，周辺のイベント（リンクによって関連しているイベント）も含めた類似度を考える．それにより，グラフ構造を考慮した類似度を定義することができた．具体的に，イベント A における文書のベクトルを  $v_A$ ，イベント B における文書のベクトルを  $v_B$  とし，イベント A およびイベント B それぞれ 1 ホップ先のイベントの文書ベクトルを集めた集合を  $S_A$  と  $S_B$  とすれば，イベント A と

イベント B の類似度は以下のように計算できる。

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

### ③ 研究課題 1 において得られた成果

#### 1) データ構造の定義

テキスト文書および UML シーケンス図を格納するデータ構造を図 3-1-1 の UML クラス図で表現する。

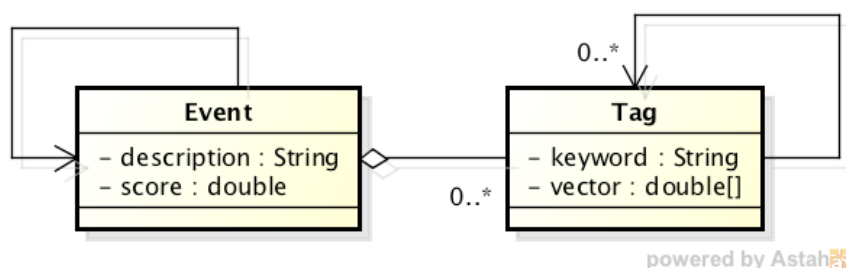


図 3-1-1 データ構造のクラス図

テキスト文書および UML シーケンスは以下の単位で分解され Event クラスを生成する。

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

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, 「スコアリング手法の調査・開発」の項目で後述) 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。さらに, 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で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), word2vec によって算出されたベクトル (vector) の属性を有する。具体的な Java ソースによるクラス定義を以下に記述する。

```

public class Tag {
    private int id;
    private String description;
    private double[] vector;
}

```

```

public class Event {
    private int id;
    private String name;
    private String description;
    private double score;
    private List<Event> link;
    private Map<Tag,Double> tags;
}

```

description から自動的に日本語の単語を抽出するためには「わかち書き」をする必要がある。これには文章における品詞を分析する形態素解析の技術を要する。例えば「にわにはにわにわとりがいる」という文章に対し、文書を解析し「にわ（庭） には にわ（二羽） にわとり（ニワトリ） がいる」という日本語として最も適切な文章になるよう品詞毎にスペースを挿入した表記から、単語として「庭」、「二羽」、「ニワトリ」を抽出する。わかち書きを実現するために、本研究では MeCab [6] の利用を行った。MeCab はオープンソースの形態素解析エンジンであり、日本語の形態素解析において最も頻繁に利用されているソフトウェアの一つである。MeCab は C/C++ 言語をもとにして記述されているが、SWIG を用いて様々な言語に対するバインディングを実装している。本研究では MeCab を description からの単語抽出に用い、後述する word2vec の学習データ作成に利用している。

## 2) 類似検索アルゴリズム

Event クラス間の類似度を算出するために、本研究では後述するように、word2vec によるタグの数値ベクトル化を行った。数値ベクトル化された Tag クラスをもとに、Event クラスの数値ベクトルは Tag クラスの TF-IDF 値をもとに以下の計算で求められる。

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

ここで、 $\text{tfidf}_{ij}$  は Tag クラス  $i$  の Event クラス  $j$  における TF-IDF 値である。また、 $w_i$  は Tag クラス  $i$  のベクトルを表す。具体的な Java ソースコードは次の通りである。

```

public class Event {
    ...
    public double[] getDocumentVector() {
        double[] vec = new double [dim];
        for (Map.Entry<Tag,Double> e : tags.entrySet()) {
            double[] tag_vec = e.getKey().getVector();
            for (int i=0; i<dim; i++) {
                vec[i] += e.getValue() * tag_vec[i];
            }
        }
    }
}

```

```

    }
    return vec;
}
}

```

ここで、Tag クラスの `getVector` メソッドは `word2vec` によるベクトルを取得するメソッドを、`dim` は予め設定されたベクトルの次元をそれぞれ表す。一方で、数値ベクトル化されたタグに対して以下の式で定義されるコサイン類似度を用いた。

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

ここで、 $S_A^n$ ,  $S_B^n$  は Event クラス A および B を中心に  $n$  ホップ先までのイベントを含めた集合を表す。具体的に以下の Java コードによる計算を行う。

```

double similarity(Set<Event> eventa, Set<Event> eventb) {
    double[] vec_a = new double [dim];
    for (Event a : event) {
        double[] tmp = a.getDocumentVector();
        for (int i=0; i<dim; i++) {
            vec_a[i] += tmp[i];
        }
    }
    double[] vec_b = new double [dim];
    for (Event a : event) {
        double[] tmp = a.getDocumentVector();
        for (int i=0; i<dim; i++) {
            vec_b[i] += tmp[i];
        }
    }
    double xy = 0.0;
    double x2 = 0.0;
    double y2 = 0.0;
    for (int i=0; i<dim; i++) {
        xy += vec_a[i] * vec_b[i];
        x2 += vec_a[i] * vec_a[i];
        y2 += vec_b[i] * vec_b[i];
    }
    return xy / (Math.sqrt(x2) * Math.sqrt(y2));
}

```

### 3.1.3 発生した問題および今後の展望

#### (1) 発生した問題

MeCab による形態素解析では、単語とその品詞を指定する辞書にもとづいて解析が行われる。MeCab では一般的に IPA 品詞体系で構築された IPADIC と呼ばれる辞書を用いる。しかしながら、分析した文書がソフトウェア設計に関するものであったため、IPADIC に登録されていない専門の単語や、特定の会社でしか使われない略語などを含んでいたことが問題点として浮上してきた。そのため、形態素解析の精度が低下するという現象が起こった。特に後述する有効性検証では Wikipedia における見出し語をすべて登録した辞書を作成したが、やはり専門性が高い単語や略語などの欠落が生じた。これは、word2vec 用の学習データ作成および Event クラスの description から単語抽出のどちらにも影響する問題であり、今後の対応が望まれる。

#### (2) 今後の展望

辞書の整備については、いくつかの階層で行う必要があると考えられる。まず、一般的な単語については今回の実験で Wikipedia からの抽出を行い、比較的高い精度で解析できることがわかった。次に領域固有の専門用語については、例えば車載システムを扱うメーカーによるコンソーシアム、IPA のような行政法人や NPO、あるいは標準化団体によって辞書が整備されることが望ましい。さらに、社内で独自に利用する略語などは、それぞれの会社において独自の辞書を作成する必要がある。

今回作成したツールでは、一般的な単語だけを整理した辞書を利用したが、上記のようにそれぞれの階層で整備された複数の辞書を統合した上で形態素解析することが、より分析精度を高めるためには望ましいものと考えられる。

## 3.2 研究課題2「スコアリング手法の調査・開発」

### 3.2.1 当初の想定

#### (1) 研究内容

安全性に着目し、FMEA や HAZOP による安全性分析結果をもとに、UML シーケンス上のシステム異常状態についての重要度をスコアリングするアルゴリズムを検討する。具体的には、過去の設計書やリスク分析結果を何らかの手法（例えば、ベイジアンフィルタ、サポートベクターマシン（SVM）などの判別分析）によって分類する。それをもとに、新しい設計上の単語から異常状態を判別するアルゴリズムを構築する。また、イベント間の関連を加味した上でマルコフ決定過程によるスコアリングアルゴリズム [7] の応用を考える。

#### (2) 想定問題と対応策

既存のスコアリングアルゴリズムを調査するだけでなく、本研究テーマの特徴を生かした新しいスコアリングの方法を開発する必要性が生じることが考えられる。その際、最新の学習理論を援用したスコアリングアルゴリズムの性能評価を行うための予備実験を合わせて行う必要がある。

### 3.2.2 研究プロセスと成果

#### (1) 研究プロセス

研究は以下の作業項目からなる。

- ・判別分析の調査・開発
- ・判別分析の実装
- ・スコアリング手法の調査・開発
- ・スコアリング手法の実装
- ・スコアリング手法の検証

判別分析の調査・開発では、類似検索の基礎をなす統計的な判別分析（クラスタリング、トピック分析）手法についての調査を行い、アルゴリズムの性能や適合性の検討を行う。判別分析の実装では、「ツールの開発・実装」に利用されるアルゴリズムの試験的な実装を行う。スコアリング手法の調査・開発では、HAZOP 安全性分析結果を利用して、設計図上の重要度スコアリングを行うための手法の調査と行う。また、スコアリング手法の実装および検証では、スコアリング手法の実装を行い、擬似的に作成したデータ（HAZOP 安全性分析結果と UML 設計図）を用いて本研究課題に適するかどうかの検証を行う。

#### (2) 具体的な研究成果の内容

##### ① 判別分析の調査・開発

判別分析の調査・開発では、まず、文書（イベント）を分類するための判別分析について調査した。判別分析とはデータをグループに分類するためのモデルを同定する手法であり、そのモデルが得られると新しいデータがどのグループに属するか判別できる。これにより、イベントのトピック分類を行うことを目標とした。最もよく知られた判別分析としてロジスティック回帰 [8] による判別分析がある。ロジスティック分析とは、ある複数の入力変

数（説明変数）と 0-1 変数を入力する確率をロジットと呼ばれる関数を用いて線形モデルとして表現するものである．具体的に，説明変数  $x_1, \dots, x_n$  と出力が 1 になる確率  $p$  の関係を

$$\log \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \dots + \beta_n x_n$$

という関係であらわし，予め与えられたデータから回帰係数と呼ばれる  $\beta_0, \beta_1, \dots, \beta_n$  を推定する．新しいデータに対する特徴量から説明変数  $x_1, \dots, x_n$  が得られると，確率  $p$  を算出することができ，0 または 1 のどちらのグループに対する近さの度合いを評価できる．

また，文書に対するトピック分類では LDA [2] がよく使われていた．LDA は確率的トピックモデルに対してベイズ推定を適用した手法として解釈される．確率的トピックモデルではトピックと単語を多項分布によって関連づける．つまり，あるトピック（「政治」，「経済」など）に属する文章は，特定の単語が特定の頻度で現れることを仮定し，その頻度を多項分布によって表現する．さらに，文章がどのトピックに属するかを確率的に表現する．実際に単語頻度を測定した文書データから，文章がどのトピックに属するかの確率（トピック分類確率）や，あるトピックに属するときの単語頻度の分布のパラメータを推定することで新たに得られた文書がどのトピックに分類されるかの確率が計算できる．このパラメータ推定にはベイズ推定が用いられている．ベイズ推定とは，あるパラメータ  $\theta$  に対する事前情報が確率分布  $p(\theta)$  で与えられるとき（事前分布），データ観測  $D$  によってパラメータ  $\theta$  に関する情報が変化した確率分布である事後分布  $p(\theta|D)$  を算出する統計的推論手続きである．事後分布の更新はベイズ則と呼ばれる以下の式で得られる．

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{\int p(D|\theta)p(\theta)d\theta}$$

上記の式において，右辺の分母は事後分布を  $\theta$  について積分した際に 1 になるようにするための定数であり正規化定数とよばれる．一般に， $\theta$  が多次元のベクトルとなるため，正規化定数の計算が困難になる場合が多い．そこで，近似的に事後分布を算出するための手法として MCMC (Markov chain Monte Carlo) 法や変分ベイズがある．LDA では変分ベイズによる計算を行っている．変分ベイズとは，観測データに対して未観測データが定義できる確率モデルにおいて，事後分布を変分近似によって近似的に算出する手法である．変分ベイズでは共役事前分布が重要な役割を果たすが，LDA ではディリクレ分布をトピック分類確率の事前分布に仮定することで変分ベイズアルゴリズムを構成している．

## ②判別分析の実装

判別分析の実装では，上述の LDA によるトピック分類を実装し，実際に文書（イベント）の分類を試みたが以下の問題点が生じた．

1. 学習に利用できるデータ量が少ない
2. 一つのイベントに対する文章量が少なく単語頻度分布が非常に偏る

最初の問題については，一般的な新聞記事や Web 上の記事では膨大な量の文書を学習（パラメータ推定）用に利用できるが，今回対象とする安全性分析では安全性分析結果として公



開されているデータが極端に少ないため、学習の精度が十分に確保できなかった。二つ目の問題については、UML シーケンス上では通常、メッセージは「CAN 送信」などの単純な語の短い文章で構成されているため、シーケンスの構成要素一つ一つを分類するためには、圧倒的に情報量が少ない。さらに、シーケンス上には社内だけで利用される略語などがあるため、特殊な単語を含むようなメッセージが高い確率でグルーピングされるので、最終的なトピック分類精度が高まらないことがわかった。上述の問題は、似たような単語も異なる単語として扱うことに起因している。つまり、「障害」と「エラー」は同じような意味であるが、それが全く異なるものとして暗に仮定されていることが問題となる。これは膨大な量の学習データで学習することで解決されるものと考えられるが、対象とする文書量が少ないため異なるアプローチで解決する必要がある。

本研究では上記の問題を解決するために、文書（イベント）のトピックを分類するのではなく、文書（イベント）を構成する単語自体を分類し、その結果に基づいて文書（イベント）の分類を行う方針を採用した。単語の分類においても数多くの既存手法が存在するが、その中でも近年、高い精度で単語の分類が行えることが報告されている word2vec [4] と呼ばれるツールを採用した。word2vec は Google の研究者によって提案されたニューラルネットワークによる単語をベクトル化する手法を実装したツールであり、深層学習の一つの応用である。利点としては、文書をまとめた単位に予め分割する必要がなく、通常の文章をまとめて学習データとすることができ、それにもとづいて、単語の近さ（同じ文脈に現れるかどうか）をベクトル表現する。これは、ツールで利用するイベントデータだけでなく、一般の書物に記載してある文章を直接学習データとして利用することができることを意味している。実際に、実証実験においては Wikipedia から取得した一般的な文章と、IPA が発行する電子書籍の文章を学習データとして利用した。

### ③ スコアリング手法の調査・開発

スコアリング手法の調査・開発では、安全性分析の観点からのスコアリングについて検討した。スコアリングについては、上述した分類に基づいた類似度をもとにした。具体的には、過去の HAZOP 分析結果に予め重要度が付加されており、新たな設計とその類似度を算出することで、新たな設計のイベントに対するスコアリングを行った。実装および検証においては次のスコアの検討を行った。いま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)

$I(\cdot)$ : 指標関数

$$I(A) = \begin{cases} 1 & A \text{ is true} \\ 0 & A \text{ is false} \end{cases}$$

このとき、ある UML シーケンスのイベント  $i$  に対して次の三通りの重要度スコアを提案した。

方法 1 : 類似度を重みとした HAZOP スコアの総和 (HAZOP スコアの合計で正規化)

$$score(i) = \frac{\sum_k sim(i,k)score(k)}{\sum_k score(k)}.$$

方法2：類似度を重みとした HAZOP スコアの平均

$$score(i) = \frac{\sum_k sim(i,k)score(k)}{\sum_k sim(i,k)}.$$

方法3：類似度が閾値  $\theta$  以上の HAZOP スコアの平均

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

上記の三つの方法のうち，協力企業における安全性分析者の意見から，方法3を採用することとした．また，上記を単一イベントの報酬とみなして，リンク構造を考慮したマルコフ決定過程によるランキングアルゴリズム [7] による重要度スコアリングを適用した．

### ③ 研究課題2において得られた成果

#### 1) 判別分析（トピック分類）の調査・開発，実装

Tag クラスに対して word2vec によるベクトル化を行う．word2vec は Google の研究者によって提案されたニューラルネットワークによる単語をベクトル化する手法を実装したツールであり，深層学習の一つの応用である．原論文 [4] では CBOW (Continuous Bag-of-Words) と skip-gram のモデルが提案されているが，本研究では CBOW を利用する．図 3-2-1 に CBOW のニューラルネットワークの構造を示す．

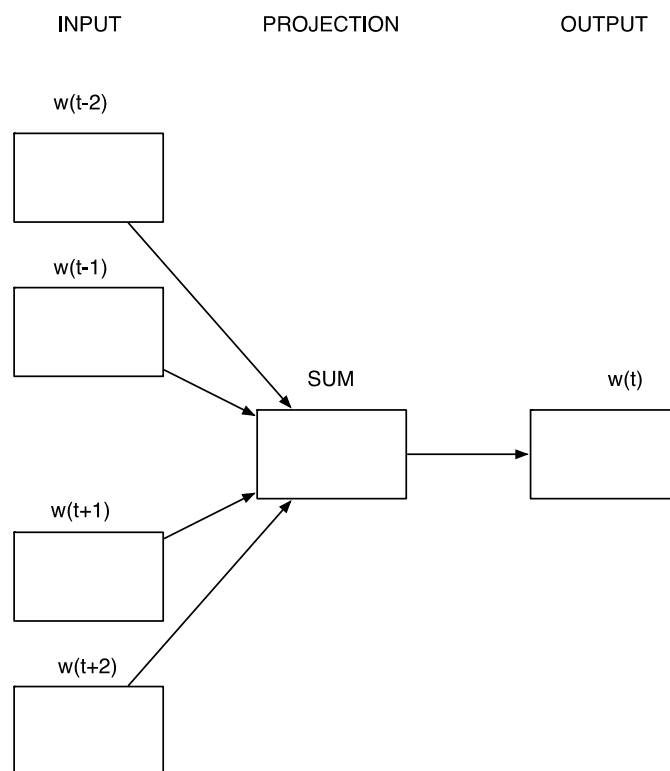


図 3-2-1 CBOW のアーキテクチャ

CBOW では周辺の単語から適切な単語を予測するよう、ニューラルネットワークによって学習する。図 3-2-1 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。本研究で開発された支援ツールでは word2vec のソースコード (<https://code.google.com/archive/p/word2vec/>) により事前に単語をベクトル化し、それを用いた類似度（前節の結果を参照）でイベント間の類似度を算出している。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。日本語コーパスには Wikipedia における文章および IPA が出版する電子書籍をテキスト化して利用した。word2vec では以下に示すいくつかのパラメータが設定でき、これらの値により精度が変化する。

train: トレーニングデータ（分かち書きデータ）

cbow: CBOW か skip-gram かの選択 cbow=1 で CBOW を利用する

size: 単語を数値化する際のベクトルのサイズ

window: 前後何文字までを単語予測に利用するか

negative: ネガティブサンプルの数

hs: 分母計算を解析的な近似 softmax にするかネガティブサンプルにするかどうかを決定する。hs=0 はネガティブサンプル

sample: ある頻度以上の文字は無視する設定値

threads: スレッド数

binary: アウトプットをバイナリにするかテキストにするか

iter: 学習の繰り返し数

min-count: min-count 以下の出現回数の単語は無視

実験においては以下に示すパラメータ設定で学習を行った。

```
word2vec -train [日本語コーパスファイル] -output vectors.txt
        -cbow 1 -size 500 -window 10 -negative 10 -hs 0 -sample 1e-5
        -threads 20 -binary 0 -iter 3 -min-count 10
```

## 2) スコアリング手法の調査・開発，実装，検証

重要度の算出は，文書（イベント）間の類似度と過去の HAZOP 分析結果に付加された重要度に基づいて行われる。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）

$I(\cdot)$ : 指標関数

$$I(A) = \begin{cases} 1 & A \text{ is true} \\ 0 & A \text{ is false} \end{cases}$$

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  $\theta$  以上の HAZOP スコアの平均

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

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 [7] を考える。この手法は Google の検索における PageRank [9, 10] の拡張として位置づけられる。PageRank は一人のユーザが Web 上のリンクをたどってランダムに巡回するとしたときに、各ページを訪れる確率によって重要度を与える手法である。このときページを訪れる確率は離散時間マルコフ連鎖の定常分布を用いて得られる。

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よう推移する。PageRank の場合と同様に、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よう (この場合はイベントを巡回する)。いま、 $\Omega$  をノード全体の集合、ノード  $i$  のリンク先を  $\Omega_i$  とする。PageRank の算出では、あるノードから他のノードへの巡回はノードの内容と関係なくすべてのノードへ等確率で移動するという仮定を設けられている。これに対して、MDP に基づく枠組みでは、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) をとるものと仮定する。すなわち、ユーザがノード  $i$  を訪問している時に選択できる行動を

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

とする。ここで、 $\Omega_i$  はリンクをたどって移動するノード、 $\text{rnd}$  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。これは、評価値を報酬とした MDP を用いてユーザのふるまいが記述できることを示唆している。

ノード  $i$  を訪問することによって得られる報酬を  $r_i$  とする。本研究では危険度に対するスコア値を報酬として定義している。さらに、ユーザが行動を決定するための評価規範を総期待割引報酬とする。これは、ユーザがノードを移動することで得られる報酬の総和である。ただし、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  $\beta^n$  を乗じてその時間的価値を減じる。最適な行動とは、少ないノード移動でより多くの報酬を得る行動を意味する。つまり、他のノードへ移動可能なハブノードとしての性質をもち、かつ周辺の危険度が高いノードへ推移することがより良い方策として選ばれる。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になる。

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。そこで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．

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

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

上記の最適性方程式を満足する  $V(\cdot)$  および  $W(\cdot)$  を見つけることで、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．具体的に最適性方程式を解くためには、Value Iteration Method [11] と呼ばれるアルゴリズムを適用する．最終的に最適な行動は  $W^a(i)$  を最大にする行動  $a$  で決定される．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．

### 3.2.3 発生した問題および今後の展望

#### (1) 発生した問題

##### ①イベント間のリンク情報について

イベント間のリンクについて、UML シーケンス図の場合は各構成要素間に関連が明確に定義されているため比較的容易に実現できる．一方で、テキスト文書に基づいた設計書や HAZOP 分析結果は必ずしもリンクが明確でない場合が多い．同時に、イベント間の類似度がイベントの粒度にも強く依存しているため、各ドキュメントにおけるイベント粒度とリンクについては対象とする問題ごとに試行錯誤的に調整する必要がある．そのためには、HAZOP 分析結果などの共通フォーマットの整備が重要になると考えられる．

##### ②word2vec の学習について

先の表であげたように word2vec は多数のパラメータが設定できるため、パラメータのチューニングが課題としてあげられる．本プロジェクトでは、一般的な場合において推奨されるパラメータ設定値を用いた．また、膨大なコーパスから学習するため、一回の学習に時間がかかることが問題点としてあげられる．そのため、本研究では高い計算能力を有する PC を必要としたことを付記しておく．

##### ③重要度定義について

重要度の定義については、ここで提案したもの以外にもいくつか考えられる．さらに類似度の定義との組み合わせを考えると、その組み合わせの種類は膨大になる．本研究では、三つの候補から実務家からの意見に応じて手法の絞り込みを行った．実際には、類似度の定義に正準相関係数を応用する手法なども検討してみたが、計算コストの観点を考慮すると採用するには至らなかった．重要度の定義は各ドメインの性質に大きく依存するため、本研究で提案したものの以外のスコアを検討する必要があることも課題として考えられる．

## (2) 今後の展望

先に指摘したように word2vec の学習は大量のデータをもとに長い計算時間を要するため、利用する PC の性能を十分に検討することが必要となる。実際の企業内で運用する場合では、word2vec 専用のサーバを構築し、リクエストされた単語のベクトルをクライアントに返すようなアーキテクチャが現実的あると考えられる。また、一般的な単語のベクトルについてはパブリックなサーバを運用し、単語ベクトル情報を随時提供できるような利用形態を実現することが考えられる。

### 3.3 研究課題3「分析例の蓄積」

#### 3.3.1 当初の想定

##### (1) 研究内容

定性的信頼性／安全性分析の精度は、リスク分析者の経験とスキルに依存する部分が大い。その問題を解決するために、本研究ではデータマイニングによる手法を用いる。データマイニングによるアプローチでは「良い分析例」（機械学習における教師データとして機能する分析方法のポートフォリオ）をデータベースに蓄積し、教師データとして利用することにより、より良い精度でリスク分析を支援することが期待される。そのため、信頼性／安全性分析の既存手法および最新手法の包括的な調査が重要となる。

##### (2) 想定問題と対応策

データマイニングで用いる教師データの質を向上するには、現行で行われている分析手法から得られる分析例だけでなく、当該分野における最新の手法を適用して得られるリスク分析事例を利用することが重要になる。そこで、最新の分析手法が発表される信頼性／安全性に関する国際会議での学術調査が重要となる。具体的には、2015 年度に開催された主要な信頼性／安全性に関する国際会議（DSN 2015, Ada-Europe 2015, COMPSAC 2015, QRS 2015, EDCC 2015, ICSME 2015, ESEM 2015, ISSRE 2015, ASE 2015, PRDC 2015, APSEC 2015）に参加し、そこで発表された最新の研究成果を調査し、本研究課題で利用可能な手法を精査する。その上で、それらの手法を適用したリスク分析事例をデータベースに蓄積し、データマイニングのための教師データとして活用する。また、国際会議では多数の有識者が参加するため、彼らとの学術的討論を通じて多角的な観点から既存の信頼性／安全性分析手法を包括的に再度調査し、これらの利用可能性を精査した上でリスク分析事例をデータベースに蓄積する。ここで作成するデータベースは「有効性検証」において利用する。

#### 3.3.2 研究プロセスと成果

##### (1) 研究プロセス

研究は以下の作業項目からなる。

- ・ 国際会議への参加と情報収集
- ・ 調査した手法による分析と結果の蓄積

国際会議への参加と情報収集では、word2vec のデータベースに蓄積できるデータが取得できるかどうかという観点から各種論文発表や報告の内容を精査する。同時に、プロジェクトで利用するデータマイニング手法や、安全性分析に直接利用できる他のリスク分析手法があるかどうかの検討を行う。調査した手法による分析と結果の蓄積では、取得した論文などの電子データをテキストデータ化し、word2vec のコーパスを作成する。また、国際会議以外でも一般的に利用できる安全性分析データが得られるかどうかの検討も行う。

##### (2) 具体的な研究成果の内容

###### ①国際会議への参加と情報収集

「DSN 2015」

会議名：DSN 2015

場所：リオデジャネイロ，ブラジル

主催：IEEE Computer Society

日程：2015 年 6 月 22 日～25 日（参加日程：2015 年 6 月 24 日～25 日）

参加できなかったワークショップに故障木分析などの発表があったが，参加したセッション全般では，本プロジェクトと直接関係する講演はなかった．本プロジェクトの定性的信頼性／安全性分析では，多角的な分析視点が必要になる．その意味では，Network Security and Consensus と Fault Tolerance で紹介された研究事例の視点は本プロジェクトでも有効となると考えられる．Network Security and Consensus では，インターネット上のセキュリティに関する発表（下記の概要を参照）があり，IoT（Internet of Things）におけるサイバーセキュリティなどのようなアプリケーション領域における事例を蓄積する必要性が認識できた．また，Fault Tolerance では，Long Latency Crash による障害の特徴付けに関する発表があり，本プロジェクトにおけるデータ整理の視点として重要であることが認識できた．

「Ada-Europe 2015」

会議名：Ada-Europe 2015

場所：ETSIT-UPM，マドリード，スペイン

主催：ACM

日程：2015 年 6 月 22 日～26 日

Ada-Europe 2015 への参加を通じて，ヨーロッパにおいて実施されている安全系システムの最新研究成果について学び，その成果を委託研究に応用できるかどうかについて分析し，詳細に検討した．具体的には IEC61508 に代表される機能安全規格への対応についての実務上の問題点や設計への対応について，主に設計段階において用いられる技術について整理した．全般的に Ada 言語に基づいたソフトウェア信頼性技術に関する実践的な応用研究が多かったが，他の会議ではあまり行く機会のない貴重な経験であった．具体的に，Critical Systems セッション，Real-time Applications セッションで紹介された実システムの安全性分析で用いられたドキュメント記述は開発システムにおいても応用することが可能であった．

「COMPSAC 2015」

会議名：COMPSAC 2015

場所：Tunghai University，台中，台湾

主催：IEEE Computer Society

日程：2015 年 7 月 1 日～5 日



COMPSAC 2015 への参加を通じて、世界各国において実施されているソフトウェア信頼性技術の最新研究成果について学び、その成果を委託研究に応用できるかどうかについて分析し、詳細に検討した。具体的には、COMPSAC Symposium on Software Engineering Technologies & Applications (7 セッション)、Stephen S. Yau Academic Symposium (3 セッション)、IEEE International Workshop on Software Test Automation (2 セッション)、IEEE International Workshop in Dependable Software & Applications (3 セッション)を聴講し、実ソフトウェア信頼性技術の全般的な研究トレンドについて整理した。具体的に、SETA1 で紹介された要求記述の手法、SETA2 と SETA3 におけるモデル記述において生じるドキュメンテーションの例示、STA1、DSA で紹介された機械学習アルゴリズムとその活用例は開発システムにおいても応用することが可能であった。

#### 「QRS 2015」

会議名： QRS 2015

場所：Marriott Hotel, バンクーバ, カナダ

主催：IEEE

日程：2015 年 8 月 3 日～5 日

QRS 2015 は QSIC と SERE というソフトウェア品質と信頼性に関する二つの国際会議が統合された国際会議であり、2015 年がその最初の会議である。ソフトウェア信頼性工学における最新研究成果について学び、その成果を委託研究に応用できるかどうかについて分析し、詳細に検討した。具体的には定量的信頼性評価やソフトウェアテスト等の最新成果に加え、上流工程における形式的方法と品質評価に関する新しい技術について調査・整理した。具体的に、Formal Modeling セッションで発表されたリスク分析結果の形式的記述方式、HSASQ Workshop に発表された UT-LDA に関する研究成果や車載ソフトウェアのテストデータ生成の際に注意する制御機能の検査は開発システムにおいても応用することが可能である。特に、本委託研究で行うトピック分析に関するいくつかの方法を知ることが出来、開発システムのエンジン部分となる形態素解析に役立った。

#### 「EDCC 2015」

会議名： EDCC 2015

場所：UPMC-Sorbonne Universites, パリ, フランス

主催：IEEE

日程：2015 年 9 月 7 日～11 日

EDCC 2015 はヨーロッパにおける高信頼化システムの主要国際会議として認知されており、2015 年より隔年開催から年次開催に変更された。高信頼化システムの設計・開発・運用に関する最新研究成果について調査し、その成果を委託研究に応用できるかどうかについて分析し、詳細に検討した。具体的には車載システムにおける信頼性に準拠した設計およびモデル化技術、ソフトウェアシステムの安全性を実現するための各種計測・評価技術について調査・整理した。具体的に、CARS Workshop, Characterization, Verification and

Validation Methods のセッションで発表された車載システムの安全性分析のためのガイドワードや仕様記述例は通常のエンタープライズ系システムと異なっており，本委託研究で行われる車載システムの安全性分析を行うために有用であった．

「ICSME 2015」

会議名： ICSME 2015

場所：Universität Bremen, ブレーメン, ドイツ

主催：IEEE

日程：2015 年 9 月 29 日～10 月 1 日

ICSME は ESEM と並ぶ実証ソフトウェア工学のトップカンファレンスであり，主に運用段階におけるソフトウェア品質評価の論文の質の高さに定評がある．本委託研究では上流工程における安全性分析に主眼を置いているため，発表された成果をそのまま直接成果に繋げることは容易でないが，本研究で予定されている企業との共同研究を通じて実証分析を行うための重要な指針を学ぶことが出来た．また，Tool Demo Track セッションでのツールデモに関する工夫はインタフェース開発に役立ち，Code Mining and Recommendation セッションで利用されたマイニング技術が有益であった．

「ESEM 2015」

会議名：ESEM 2015

場所：北京, 中国

主催：IEEE Computer Society

日程：2015 年 10 月 19 日～23 日（19 日～20 日 ISERN 2015, 21 日 Promise 2015, 2015 年 10 月 22 日～ 23 日 ESEM 2015）

ISERN 2015 への参加を通じて，世界各国の研究者の関心をわかるようになった．特にプログラム技術だけではなく，開発の各工程においても定性と定量の手法が必要であることが認識できた．また，上流工程のデータマイニングの研究が多くなかったが，定性的な手法の提案があり，この知見は本研究プロジェクトに有益であった．Promise 2015 にて，ソフトウェア欠陥に関する研究が多く紹介された．特にソースコードの“におい”に関する発想は面白く，欠陥検出と予測の手法が本研究プロジェクトに応用できる．ESEM 2015 にて，ソフトウェア工学の定量分析に関する研究が多く紹介された．特に企業から，開発現場の話があり，理論の応用状況も紹介された．本研究で予定されている企業との共同研究を通じて実証分析を行うための重要な指針を確認することが出来た．

「ISSRE 2015」

会議名：ISSRE 2015

場所：ゲイサズバーグ, 米国

主催：IEEE Computer Society

日程：2015 年 11 月 2 日～5 日

本プロジェクトに関連して、信頼性分析や安全性分析の教師データ (word2vec に入力するデータ) として利用できるかどうかという観点で調査を行った。ISSRE 2015 ではいくつかの実証研究報告 (experience report) が発表されており、その中でもバグの分類や管理に関する調査報告があり、それらをデータとして利用することができた。

「ASE 2015」

会議名：ASE 2015 場所：リンカーン, 米国

主催：ACM, IEEE Computer Society

日程：2015 年 11 月 9 日～13 日 (参加日程：2015 年 11 月 12 日)

参加できるセッションが少なかったため、本プロジェクトと直接関係する講演はなかったが、Software performance におけるプロジェクト管理における自動化では、プロジェクト管理上のデータから学習アルゴリズムを通じた自動化を行うが、そのデータ構造が本プロジェクトで扱うデータ構造と類似していたため学習の手法の観点から参考になった。

「PRDC 2015」

会議名：PRDC 2015

場所：張家界, 中国

主催：IEEE Computer Society

日程：2015 年 11 月 18 日～20 日

安全性に関する発表について手法として直接参考になるものは得られなかった。一方で、ハードウェアおよびソフトウェアの安全性・信頼性評価に関する文献を教師データ (word2vec に入力するデータ) として利用できた。

「APSEC 2015」

会議名：APSEC 2015

場所：Holiday Inn New Delhi International Airport, ニューデリー, インド

主催：IEEE

日程：2015 年 12 月 1 日～4 日

APSEC 2015 への参加を通じて、アジア地域において実施されている安全系システムの最新研究成果について学び、その成果を委託研究に応用できるかどうかについて分析し、詳細に検討した。具体的には車載システムや鉄道システムの設計手続きを通じて、開発データベースに含めるタグ情報、ガイドワード、ドキュメント情報についての調査研究を行った。ソフトウェア工学全般に関する会議であったので上流から下流にかけての幅広い分野を網羅していたので、他の会議ではあまり聞く機会のない最新の成果について学ぶことが出来た。この知見は本プロジェクトに有益であった。

## ②調査した手法による分析と結果の蓄積

国際会議で得られた論文の電子データをすべてテキストデータ化し、word2vec に対するコーパスとして利用した。さらに、標準的な分析データとして JasPar (Japan Automotive Software Platform and Architecture) で実施したソフトウェア安全要求・設計仕様書を得ることができた。また、word2vec の学習データとして以下の IPA が発行する電子書籍を入手し、word2vec の学習データ（日本語コーパス）として利用した。

### 【新版】組込みスキル標準 ETSS 概説書

組込みスキル標準 ETSS 導入推進者向けガイド

組込みスキル標準 ETSS 教育プログラムデザインガイド

組込みソフトウェア向けプロジェクトマネジメントガイド [計画書編]

【改訂版】組込みソフトウェア開発向けコーディング作法ガイド [C 言語版] Ver. 2.0

【改訂版】組込みソフトウェア開発向けコーディング作法ガイド [C 言語版] Ver. 1.1

組込みソフトウェア開発向けコーディング作法ガイド [C++言語版]

【改訂版】組込みソフトウェア開発向け品質作り込みガイド

【改訂版】組込みソフトウェア向け開発プロセスガイド

組込みソフトウェア向け設計ガイド ESDR [事例編]

組込みソフトウェア向けプロジェクト計画立案トレーニングガイド

組込みソフトウェア開発における品質向上の勧め[バグ管理手法編]

組込みソフトウェア開発における品質向上の勧め[ユーザビリティ編]

組込みソフトウェア開発における品質向上の勧め[設計モデリング編]

組込みソフトウェア開発における品質向上の勧め（コーディング編）

組込みソフトウェア開発におけるプロジェクトマネジメント導入の勧め

組込みシステムの安全性向上の勧め（機能安全編）

組込みソフトウェア開発における品質向上の勧め [テスト編～事例集～]

IT プロジェクトの「見える化」～上流工程編～

IT プロジェクトの「見える化」～中流工程編～

IT プロジェクトの「見える化」～下流工程編～

IT プロジェクトの「見える化」～総集編～

情報処理システム高信頼化教訓ハンドブック（IT サービス編）2013 年度版

情報処理システム高信頼化教訓ハンドブック（製品・制御システム編）2013 年度版

超上流から攻める IT 化の原理原則 17 ケ条

経営に活かす IT 投資の最適化 ～情報システムを安心して快適に使うために～

プロセス改善セミナー 事例紹介

非機能要求グレード利用ガイド[活用編]

高回復力システム基盤導入ガイド（概要編）

高信頼化ソフトウェアのための開発手法ガイドブック

エンピリカルソフトウェアエンジニアリングの勧め

実務に活かす IT 化の原理原則 17 ケ条

プロセス改善ナビゲーションガイド ～虎の巻編～

プロセス改善ナビゲーションガイド～ベストプラクティス編～

プロセス改善ナビゲーションガイド～なぜなに編～

プロセス改善ナビゲーションガイド～プロセス診断活用編～

プロセス改善ナビゲーションガイド～自律改善編～

定量的品質予測のススメ

続 定量的品質予測のススメ

### **3.3.3 発生した問題および今後の展望**

#### **(1) 発生した問題**

収集したドキュメントは word2vec に対する学習データと過去の分析例としてイベントへ分割した後にデータベースへ格納するものに分けられる。word2vec に対するデータはテキストデータそのままを扱うため（PDF からテキスト抽出が可能であれば）容易に学習データとして利用できる。一方、イベント分解をする必要があるデータ（UML シーケンス図以外）については現状手動でデータ入力用の XML に変換する必要がある。HAZOP 分析結果などは Microsoft Excel で記述されていることが多く、本研究では XML 化の作業を Excel の機能を使うことで工夫を行った。

#### **(2) 今後の展望**

研究課題 2 でも述べたが、ドキュメントについては自動的にリンク構造などが同定できることが望ましい。そのためには、ドキュメントフォーマットの標準化（特に HAZOP 分析などの安全性分析）が行われることで今後同様なツールを作成するための補助となり得る。

### 3.4 研究課題4「ツールの開発・実装」

#### 3.4.1 当初の想定

##### (1) 研究内容

研究課題1および研究課題2で述べた検索手法を実装したツールの開発を行う。ツールはGUIによるフロントエンドとデータベースからなる。GUIの作成にはJavaを用いる。また、GUIなどの開発は外部業者に委託する。検索手法やスコアリングに関するアルゴリズム開発については広島大学で行う。このとき、先の研究課題で作成したものをGUIとのインタフェースを考慮して再実装する。

##### (2) 想定問題と対応策

データベースに蓄えるデータは様々なものが考えられるため、MySQLなどのデータベースで格納するための形式を十分に考慮しなければならない。外部業者へ委託する部分の仕様書作成については研究プロジェクトの初期で行うため、検索効率やデータ保守性の観点からできるだけ戻り作業が生じないように心掛けた。

#### 3.4.2 開発プロセスと成果

##### (1) 開発プロセス

研究は以下の作業項目からなる。

- ・システム設計
- ・RFPの作成
- ・検索手法の再実装
- ・スコアリング手法の再実装

システム設計では広島大学側でツール全体のアーキテクチャに対する設計を行い、外部委託業者へ発注する部分の明確化とインタフェースなどの設計を行う。RFPの作成では、外部委託業者へ発注するためのライブラリの要件定義書を作成する。検索手法およびスコアリング手法の再実装では、研究課題1および研究課題2で作成した試作版を処理性能なども考慮した再度実装（リファクタリング）を行う。

##### (2) 開発内容

###### ①システム設計, RFPの作成

ツール開発の大まかなシステム設計は研究プロジェクトの初期段階で行い、RFP (Request For Proporsal) を作成した。外部業者の選定は広島大学の規則に則った入札にもとづいて行われた。作成したRFPは以下の通りとなった。

---

## II 技術的要件

(性能, 機能以外に関する要件)

### 1. 開発プロセス

#### 1-1 設計

(1) 基本設計（アーキテクチャ設計）および詳細設計については担当者によるレビューを受けること。

(2) 最終的な基本設計書および詳細設計書を納品すること。

#### 1-2 単体テスト

(1) Java クラス単位および DB のテーブル単位に対するテストの項目については受注者の責任において選定および実施すること。

(2) 単体テスト項目およびその検証結果の詳細を納品する必要はないが、単体テストの概略（テスト範囲および実施工数など）についてはテスト報告書に記載した上で、納品すること。

#### 1-3 結合テスト

(1) 結合テストに対するテストの項目については受注者の責任において選定および実施すること。

(2) 検索エンジンとのインタフェース部分については担当者と協議を行った上でテスト項目を決定する。

(3) 結合テストで実行するすべてのテスト項目および検証結果（不具合が検出された場合は、修正と回帰テスト実施による検証結果などのバグトラッキング情報を含む）はテスト報告書として納品すること。

#### 1-4 システムテスト

(1) システムテストに対するテストの項目については受注者の責任において選定および実施すること。

(2) 検索エンジンとのインタフェースについては担当者と協議を行った上でテスト項目を決定する。

(3) システムテストで実行するすべてのテスト項目および検証結果（不具合が検出された場合は、修正と回帰テスト実施による検証結果などのバグトラッキング情報を含む）はテスト報告書として納品すること。

(4) 開発するシステムは、以下の環境における稼働を想定としているため、すべての環境に対するシステムテストを行うこと。

- Java VM の動作環境：Windows 7～, Mac OS X 10.9～
- DBMS の動作環境：Windows 7～, Mac OS X 10.9～

(5) テスト報告書には受注者によるシステムの信頼性評価とその根拠（手法は問わない）を含めること。

#### 1-5 受入テスト

(1) 動作させる環境においてテストを実施すること。

(2) テスト項目については担当者と協議の上で決定する。

(3) 受入テストで実行するすべてのテスト項目および検証結果（不具合が検出された場合は、修正と回帰テスト実施による検証結果などのバグトラッキング情報を含む）はテスト報

告書として納品すること。

## 2. 保守

- (1) 納品完了後の保守については必要ない。
- (2) 納品完了は本番環境において実施する受入テスト結果を受けて、担当者と協議の上で判定を行う。

(性能、機能に関する要件)

### 1. 構成

- (1) 調達するシステムは以下の DBMS と二つのモジュールからなる。
  - DBMS : オブジェクトを格納する DBMS. イベントオブジェクトを管理する DB (イベント DB) とタグオブジェクトを管理する (タグ DB) がある。
  - DB 管理モジュール (db.jar) : イベント DB, タグ DB を管理するモジュール
  - レンダリングモジュール (render.jar) : 結果を描画するモジュール
- (2) モジュールと DBMS 間はクライアントサーバモデルで設計すること. 納品するシステムでは DBMS とモジュールが同じマシン上で動作しても良いが, ネットワークを介したシステムへの移行が容易であること。

### 2. 要素技術

- (1) 独自のソフトウェアを DBMS に利用する場合は, 将来のメンテナンスを考え, そのソフトウェアのソースを公開可能であること。
- (2) 利用するソフトウェアは, 実在するシステムで多くの稼働実績があること。

### 3. システム構成要素の詳細

#### A. DBMS

##### A-1 DBMS の基本要件

- (1) A-2, A-3 で示される Event クラスのインスタンス (Event オブジェクト) および Tag クラスのインスタンス (Tag オブジェクト) を保持する。
- (2) 「DB 管理モジュール」提供するインタフェースにもとづいて適切な DBMS の選択およびテーブルの設計を行うこと。
- (3) Java で実装される「DB 管理モジュール」を介して, 所望のインスタンスを取得できる仕組み (テーブル設計なども含めて) を基本設計および詳細設計で明示すること。
- (4) 作成するシステムのパフォーマンスの目安を以下の状況それぞれで提示すること。
  - 10,000 件程度の Event オブジェクトが格納されている状況において, タグによるイベントの検索を行った場合のレスポンスタイム
  - 10,000 件程度の Event オブジェクトが格納されている状況において, テキストによるイベントの全文検索を行った場合のレスポンスタイム
  - 10,000 件程度の Tag オブジェクトが格納されている状況において, タグによるタグオブジェクトの検索を行った場合のレスポンスタイム
- (5) DBMS のアクセス権管理については, 納品時に設定する必要はない。納品後, 適切に



設定できるようにすること.

(6) データのバックアップが納品後に適切に行えること.

## A-2 Event クラス

(1) 以下のフィールド変数を有するクラスとして設計すること.

```
/**
 * Event クラス
 *
 * @author HiroshimaUniv.
 * @version 1.0
 */

public class Event {
    /**
     * 時系列的に前のイベント (リスト)
     */
    List<Event> prev;

    /**
     * 時系列的に次のイベント (リスト)
     */
    List<Event> next;

    /**
     * テキストデータによる詳細
     */
    final String description;

    /**
     * タグ (イベントの特徴的なキーワード)
     */
    List<Tag> tags;
}
```

(2) prev, next を外部から List として取得するメソッドを有すること.

(3) prev, next を外部から変更 (追加と削除) するための手段を有すること.

(4) tags を外部から List として取得するメソッドを有すること.

(5) tags を外部から変更 (追加と削除) するための手段を有すること.

(6) description に対する getter を有すること.

### A-3 Tag クラス

(1) 以下のフィールド変数を有するクラスとして設計すること.

```
/**
 * Tag クラス
 * @author HiroshimaUniv.
 * @version 1.0
 */

public class Tag {
    /**
     * テキストデータによるキーワード
     */
    final String keyword;

    /**
     * 近いキーワードとその類似度
     */
    Map<Tag,Double> nearestNeighbors;

    /**
     * 遠いキーワードとその類似度
     */
    Map<Tag,Double> reverseNearestNeighbors;
}
```

(2) nearestNeighbors, reverseNearestNeighbors を外部から Map として取得するメソッドを有すること.

(3) nearestNeighbors, reverseNearestNeighbors を外部から変更 (追加と削除) するための手段を有すること.

(4) keyword に対する getter を有すること.

## B. DB 管理モジュール

### B-1 基本要件

(1) DB 管理モジュールは以下で構成される jar ファイルとなる.

- 「A-2 Event クラス」で示した Event クラスの Java による実装.
- 「A-3 Tag クラス」で示した Tag クラスの Java による実装.
- イベントオブジェクトを管理する EventDBMgr クラスの Java による実装.

- タグオブジェクトを管理する TagDBMgr クラスの Java による実装.
- (2) EventDBMgr は以下の機能を有する.
- 与えられた Tag クラスを含むすべての Event クラスの検索・抽出.
  - 与えられた keyword に一致する Tag クラスを含むすべての Event クラスの検索・抽出.
  - テキストデータで与えられた検索条件にマッチする description を持つ Event クラスの検索・抽出 (description に対する全文検索).
  - Event クラスの生成・削除
- (3) TagDBMgr は以下の機能を有する.
- 与えられた Tag クラスを nearestNeighbors として含むすべての Tag クラスの検索・抽出.
  - 与えられた Tag クラスを reverseNearestNeighbors として含むすべての Tag クラスの検索・抽出.
  - 与えられた keyword に一致する Tag クラスを抽出.
  - Tag クラスの生成・削除

#### A-2 EventDBMgr インタフェース

- (1) EventDBMgr は以下のインタフェースを持つ.

```
public interface EventDBMgr {
    Set<Event> searchWithTag(Tag tag);
    Set<Event> searchWithTag(String keyword);
    Set<Event> searchWithQuery(String query);
    Event create(String description,
        List<Event> prev,
        List<Event> next,
        List<Tag> tags);
    void delete(Event event);
}
```

- (2) 各メソッドは以下の通り.

メソッド	説明
Set searchWithTag(Tag tag)	与えられた tag を含むすべての Event オブジェクトを抽出した Set を返す.
Set searchWithTag(String keyword)	与えられた keyword に対応する Tag オブジェクトを含むすべての Event オブジェクトを抽出した Set を返す.
Set searchWithQuery(String query)	与えられた検索条件 query にマッチする description を持つすべての Event オブジェ

	クトを抽出した Set を返す. query は AND 条件, OR 条件, NOT 条件が扱えるキーワード検索を表すテキストデータとして与えられる (例: AND 検索の例として「速度 & 危険」のテキストデータ (query) は「速度」と「危険」を description 内に含む Event オブジェクトを返す).
Event create(String description, List prev, List next, List tags)	prev, next, tags, description を要素とする新しい Event オブジェクトを生成する (データベースに登録する).
void delete(Event event)	event に対応する Event オブジェクトをデータベースから削除する. 該当する event が別の Event オブジェクトの prev, next に格納されている場合はリストから取り除く.

### B-3 TagDBMgr インタフェース

(1) TagDBMgr は以下のインタフェースを持つ.

```
public interface TagDBMgr {
    Map<Tag,Double> searchNearestTag(Tag tag);
    Map<Tag,Double> searchNearestTag(String keyword);
    Map<Tag,Double> searchReverseNearestTag(Tag tag);
    Map<Tag,Double> searchReverseNearestTag(String keyword);
    Tag getTag(String keyword);
    Tag create(String keyword,
        Map<Tag,Double> nearestNeighbors,
        Map<Tag,Double> reverseNearestNeighbors);
    void delete(Tag tag);
}
```

(2) 各メソッドは以下の通り.

メソッド	説明
Map searchNearestTag(Tag tag)	与えられた tag を近いタグとして含むすべての Tag オブジェクトとその類似度を抽出した Map を返す.
Map searchNearestTag(String keyword)	与えられた keyword に対する Tag オブジェクトを近いタグとして含むすべての Tag オブジェクトとその類似度を抽出した Map を返す.
Map searchReverseNearestTag(Tag tag)	与えられた tag を遠いタグとして含むすべての

tag)	の Tag オブジェクトとその類似度を抽出した Map を返す.
Map searchReverseNearestTag(String keyword)	与えられた keyword に対する Tag オブジェクトを遠いタグとして含むすべての Tag オブジェクトとその類似度を抽出した Map を返す.
Tag getTag(String keyword)	与えられた keyword に対する Tag オブジェクトを返す.
Tag create(String keyword, Map nearestNeighbors, Map reverseNearestNeighbors)	keyword, nearestNeighbors, reverseNearestNeighbors を要素とする新しい Tag オブジェクトを生成する (データベースに登録する).
void delete(Tag tag)	tag に対応する Tag オブジェクトをデータベースから削除する. 該当する tag が別の Tag オブジェクトの nearestNeighbors, reverseNearestNeighbors にある場合はリストから取り除く.

### C. レンダリングモジュール

#### C-1 基本要件

- (1) 与えられたイベント集合 (Set や Map) から結果を GUI で表示する java クラスで構成される jar ファイルとなる.
- (2) Java Swing のコンポーネント (例えば JPanel クラスなど) を継承したクラスを生成するメソッドを有すること.
- (3) 基本の表示形式として以下の二つが可能であること.
  - ダイアグラム表示 (ノードと矢印を用いた表示)
  - リスト表示 (表形式)
- (4) Microsoft Excel 形式によるレポートの出力が可能であること

#### C-2 ダイアグラム表示

- (1) 与えられたイベント集合 (Set) に格納されている Event オブジェクトの prev, next の情報からダイアグラムを表示する Java オブジェクトを生成する.
  - イベントをノード, イベント間の時系列的な前後関係 (prev, next の情報) をノード間の矢印で表現するダイアグラムを表示する. ただし, 与えられたイベント集合に格納されているイベントとそれ以外のイベントはノードの色を変えることによって視覚的に区別できるようにすること.
  - イベントノードをクリック (もしくは OnMouse) することでイベントの内容 (description) が別コンポーネントに表示されること.
  - ダイアグラム表示では, 拡大縮小やドラッグによる表示範囲の移動ができるよう

にすること.

(2) 描画を行う Java オブジェクトは Swing の描画を行うオブジェクトの派生クラスであり, 担当者による表示のカスタマイズが容易に行えること.

### C-3 リスト表示

(1) 与えられたイベント集合とその重要度 (Map) から, 集合の要素となっている Event オブジェクトを重要度の高い順番にソートして, 一つの行が一つのイベントに対応する表を表示する Java オブジェクトを生成する.

- イベントをクリックすることでイベントの内容 (description) が別のコンポーネントに表示される.
- ドラッグ&ドロップあるいはカット&ペーストなどの操作で行を移動させることによりイベントの表示順が変更できること.
- イベントの表示順を変更後, その順番に関する情報を取得するメソッドを有すること.

(2) リスト表示を行う Java オブジェクトは Swing のオブジェクトの派生クラスであり, 担当者による表示のカスタマイズが容易に行えること.

### C-4 レポート出力機能

(1) C-3 のリスト表示の結果に準じたレポートを Microsoft Excel 形式で出力できること.

(2) レポートの出力内容の詳細は開発段階においてその仕様変更に対応できること.

---

## ② 検索手法の再実装, スコアリング手法の再実装

広島大学側で作成する検索エンジン部分については, 前述の研究課題の結果から Java により再実装を行った. なお, 上述の RFP から以下の変更点があった.

- Event クラスのフィールド変更 スコア値などを納める必要が生じたため, それらを格納するフィールドを追加した.
- Tag クラスのフィールド変更 word2vec によるベクトル値を格納する必要が生じたため, そのフィールドを追加した.

上記の作業により作成したツールのアーキテクチャおよび機能を図 3-4-1 に示す.

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」の機能を有する. 作成したツール全体のアーキテクチャは以下の通り. 赤で示している部分は外部業者による開発を表す.

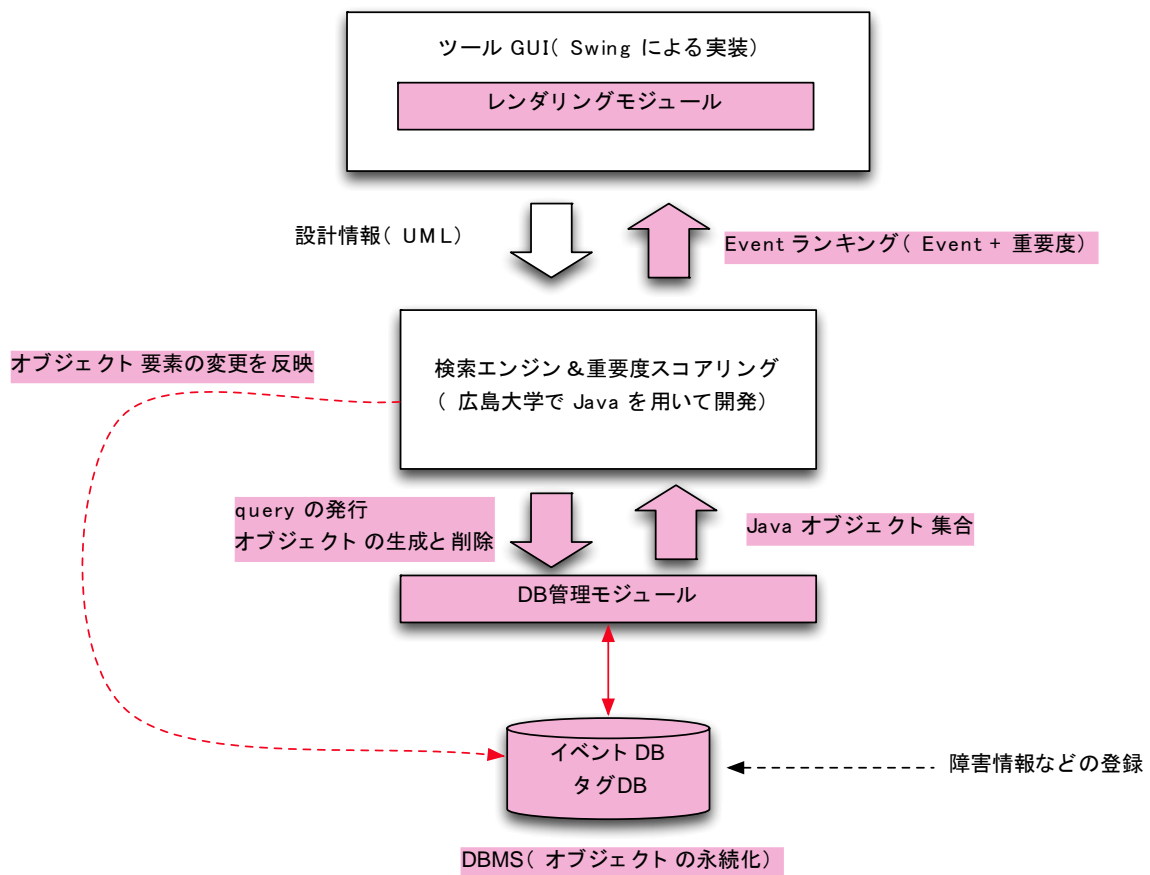


図 3-4-1 システムアーキテクチャ

また，Java クラス (Event クラス，Tag クラス) の永続化には sqlite によるデータベースを用いた。

「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．図 3-4-2 にタグ登録画面を示す．実証実験においては，Wikipedia，IPA が発行する電子書籍，国際会議にて収集したデータのテキストを利用し，667644 単語に対するベクトル化を行っている．また，バイナリ形式にすることにより，667644 単語のベクトルデータを 2.68GB のファイルサイズで実現できている．

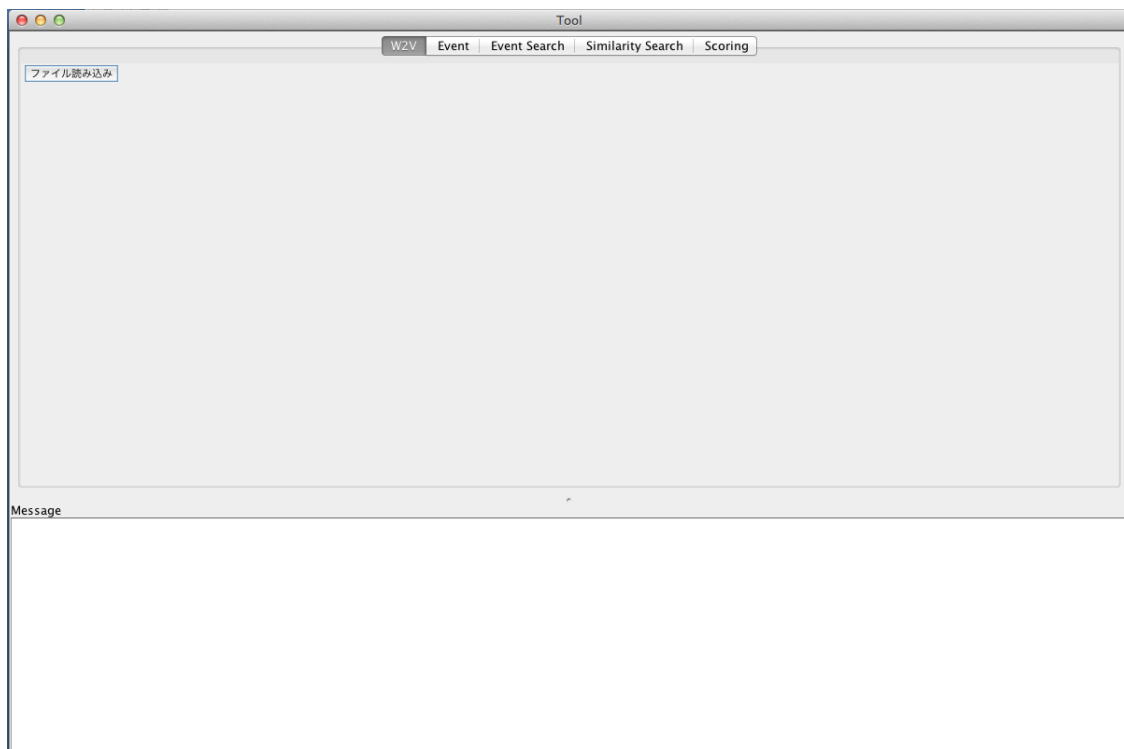


図 3-4-2 タグ DB 登録画面

「イベント DB 登録」では XML 化されたドキュメントまたは EA (Enterprise Architecture) 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。ドキュメントに対するイベント登録用の XML は以下ようになる。

```
<?xml version="1.0" encoding="UTF-8"?>
  <RootEvent>
    <Event ID="イベント名">
      <Description>イベントに対する文章</Description>
      <Score>重要度スコア（省略可）</Score>
      <Links>
        <Link>関連するイベントの Event ID</Link>
        <Link>関連するイベントの Event ID</Link>
      </Links>
    </Event>
    ...
  </RootEvent>
</xml>
```

我が国では、HAZOP 分析結果をはじめとするドキュメント類は Microsoft Excel で管理されることが多い。そこで、Excel のセルを上記の、ID, Description, Score, Links の四つの列で整理し、そのシートから生成されるタブ区切りのテキストファイルから上記の XML



を生成するために、 Ruby 言語を用いた以下のスクリプトを作成した.

```
#!/usr/bin/ruby
require 'set'

class Event
  def initialize(id, desc, score)
    @id = id
    @desc = desc
    @score = score
    @links = Set.new
  end

  def disp
    puts "<Event ID=#{@id}>"
    puts "<Description>#{@desc}</Description>"
    puts "<Score>#{@score.to_f}</Score>"
    puts "<Links>"
    @links.each do |x|
      puts "<Link>#{x}</Link>"
    end
    puts "</Links>"
    puts "</Event>"
  end

  def addLink(dest)
    @links.add(dest)
  end
end

h = Hash.new
alllink = []

while x=gets do
  a = x.split(/\t/)
  id = a[0].gsub(/['']/, " ")
  desc = a[1].gsub(/['']/, " ")
  score = a[2]
  event = Event.new(id, desc, score)
  h[id] = event
end
```

```

links = a[3].chop.gsub(/['']/, " ")
links.split(/,/).each do |x|
  alllink.push([id,x])
end
end

alllink.each do |link|
  if h[link[0]] != nil
    h[link[0]].addLink(link[1])
  end
  if h[link[1]] != nil
    h[link[1]].addLink(link[0])
  end
end

puts "<?xml version=¥"1.0¥" encoding=¥"UTF-8¥"?>"
puts "<RootEvent>"
h.each do |key,value|
  value.disp
end
puts "</RootEvent>"

```

一方、シーケンス図は EA がエクスポートする XML ファイルを直接解析して UML コンポーネントおよびそれらのリンクを登録できる。

「類似度検索」では、まず検索をするもとのイベントを、登録してあるイベントから選択する（図 3-4-3 を参照）。次に指定した検索範囲に該当するイベントすべてに対する類似度を算出する（図 3-4-4 を参照）。検索範囲を省略した場合、登録しているすべてのイベントに対する類似度を算出する。

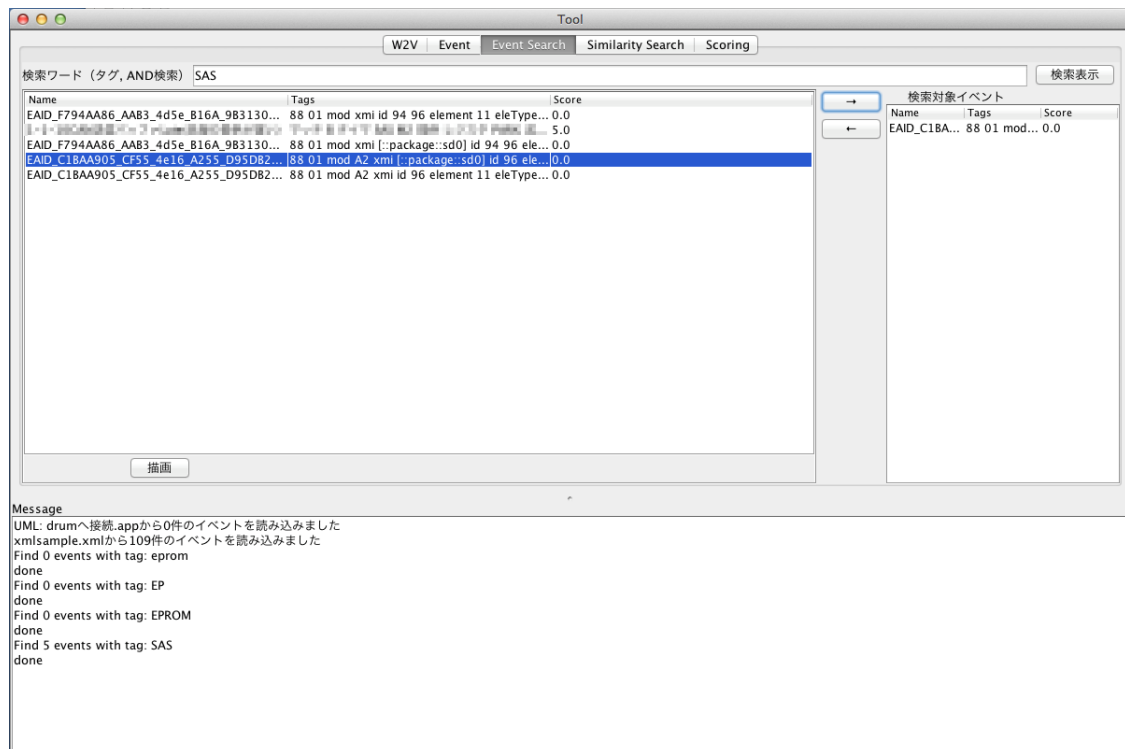


図 3-4-3 類似度検索（検索をするもとのイベントを設定）

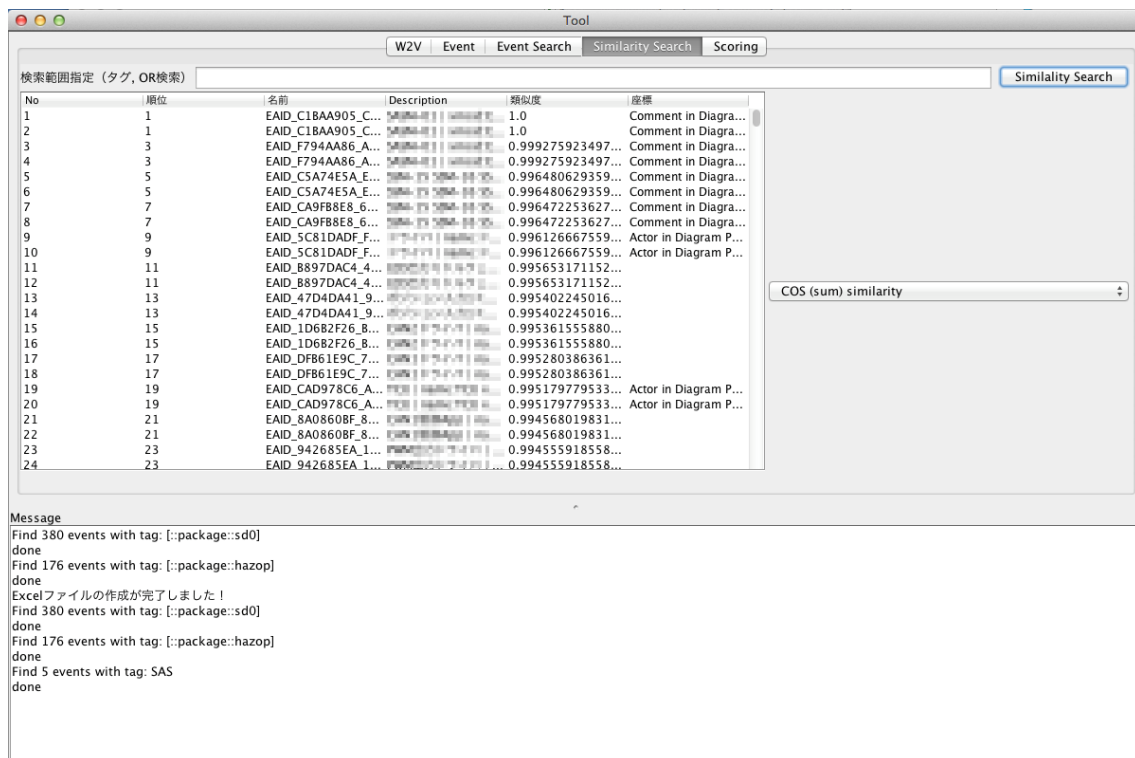


図 3-4-4 類似度検索（検索結果）

「重要度スコアリング」では、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。図 3-4-5 に重要度スコアリングの画面を示す。スコアリング手法には類似度を閾値として用いるもの（「研究課題 2」を参照）と MDP によるもののそれぞれ二種類を適用できるように設定した。また類似度にはコサイン類似度 (COS similarity) およ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) の二種類が利用できるようにした。

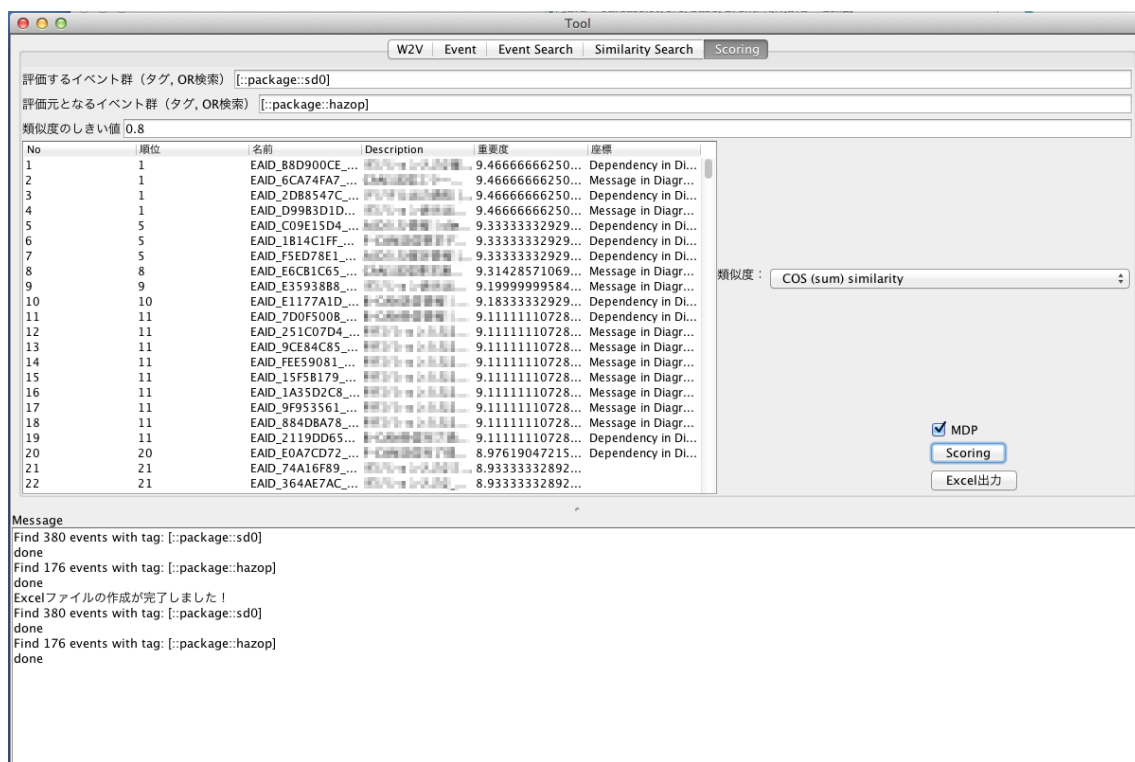


図 3-4-5 重要度スコアリング

### 3. 4. 3 発生した問題および今後の展望

#### (1) 発生した問題と対処

RFP から Event および Tag クラスについて修正があった。大幅な戻り作業を防ぐために、最初に設定したフィールドで対応できるようにした。例えば、Event クラスのリンク情報については、当初、先のイベントと後のイベントをリンクする構造を想定していたが、その一方だけを使うことで、初期の DB 設計そのままで利用できるようにした。しかしながら、フィールド名と情報の意味が食い違っている部分が生じてくる場合があるので、将来的にはリファクタリングが必要になることを指摘しておく。

また、DB の仕様についても、当初は外部委託業者から MySQL の利用が提案されていたが、ツールのポータビリティや企業への導入の容易性などを考慮して sqlite に変更した。この変更では、MySQL も sqlite も SQL によるクエリーに大きな違いがなかったため大きな戻り作業は発生しなかった。

## **(2) 今後の展望**

結果の出力については、図 3-4-5 に示すようなリスト形式になっている。研究課題 5 の協力会社からの意見でも挙げられているが、出力については UML シーケンスと対応がとれるような出力の工夫が今後は必要になる。

### 3.5 研究課題5「有効性検証」

#### 3.5.1 当初の想定

##### (1) 研究内容

ツールの有効性は類似検索の精度に強く依存しており、類似検索の精度はデータベースに蓄えられるデータの質や量に依存する。そこで、「分析例の蓄積」で作成されたデータベースと、実際の企業における安全性分析者が行ったリスク分析から作成される結果との比較を行い、本研究で開発したツールによる支援の有効性を検証する。また、データベースの質による支援の精度を検証するため、実際の企業における安全性分析者が行ったリスク分析例だけから成るデータベースも別途構築し、「分析例の蓄積」で作成されたデータベースとの比較を行う。

##### (2) 想定問題と対応策

企業において、定性的な分析の結果は機密事項である。機密保持のため、実際に市場へ出荷する実システムの分析結果を直接用いることができない場合が想定される。そのような場合は、企業で試験的に行った信頼性／安全性分析の結果を用いることも検討する。また、機密保持契約を結んだ上で企業との協力体制が不可欠となる。

#### 3.5.2 研究プロセスと成果

##### (1) 研究プロセス

研究は以下の作業項目からなる。

- ・ 協力企業の選定
- ・ 分析結果のデータベース化
- ・ ツールを用いた分析
- ・ 従来手法との比較

協力企業の選定では、データの提供および結果の検証を行ってもらう企業の選定を行う。分析結果のデータベース化では提供されたデータのデータベース化を行う。ツールを用いた分析および従来手法との比較では、ツールによる分析結果の提示と、従来からそれぞれの企業で行われている安全性分析結果を比較、および標準的なデータだけを利用した場合との比較を通じてツールの有効性を検証する。

##### (2) 具体的な研究成果の内容

###### ① 協力企業の選定、分析結果のデータベース化

協力企業として、主に車載システム開発に携わる企業（A社）と空調制御システムを開発する企業（B社）と機密保持契約を結び、リスク分析と開発システムの有効性検証を行った。まず、各社から製品の設計情報、特に制御時のシーケンス図を受け取りイベントDB上へ登録を行った。また、A社からは安全性分析結果（HAZOP分析結果）もデータとして得ることができた。

###### ② ツールを用いた分析、従来手法との比較

A 社、B 社いずれの検証においても、HAZOP 分析結果（安全性分析結果）をもとにして、シーケンス上の重要な部分をスコアリングし重要度によりランク付けを行った。それを結果として提示し、各企業における開発担当者からの意見を収集した。スコアリングのもととなる HAZOP 分析結果は、A 社の場合では二種類（社内の安全性分析者が行ったものと、研究課題 3 で得られた JasPar によって提示された例）のものをそれぞれ用いた。特に、JasPar の結果を使った理由は、異なるアプリケーションドメインの結果が利用できるかどうかの検証を行うためであった。また、B 社の場合は、同じようなアプリケーションドメインの安全性分析結果が得られなかったため、JasPar のもののみを利用した。

上記の手順によって得られた成果を協力企業毎にまとめる。

#### 1) 実証分析（A 社）

##### a. 利用データ

利用する設計データは、自動車のシフト制御に関するシーケンス図であり、二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う。重要度スコアリングのもととなる 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）で実施されたもの（HAZOP1）と、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）で比較を行った。

シーケンス図をイベントに分解して登録した際のイベント数は SD0, SD1 とともに 380 イベントであった。また、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以下の 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。この分析結果には、安全性分析者によって予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）で評価してもらった。

表 3-5-1 HAZOP ガイドワード

タイプ	ガイドワード	説明
ネガティブ	NO (NOT)	意図した事象、出力が起こらない
量的変更	MORE	意図した値の範囲より大きい
	LESS	意図した値の範囲より小さい
質的変更	AS WELL AS	似て非なる事象、値が発生する
	PART OF	データあるいは信号が不完全である
代替	REVERSE	逆相、逆符号など
	OTHER THAN	不正確、もしくは意図とは異なる事象が発生する
時間	EARLY	意図より時間的に早く事象が発生する
	LATE	意図より時間的に遅く事象が発生する
順序	BEFORE	意図より早い順序で事象が発生する
	AFTER	意図より遅い順序で事象が発生する

HAZOP1 では一つの障害シナリオを一つのイベントと設定し、合計 176 のイベントを DB に

登録した．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，同様に表 3-5-1 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．登録したイベント数は 91 である．重要度については ASIL のレベルに応じた 5 段階の評価を行った．

#### b. 分析結果と考察

HAZOP1 を使った評価では，シーケンス SD0, SD1 とともに類似度の閾値 0.8 として評価を行った．重要度は類似度の組み合わせと MDP に基づいたアルゴリズムの利用を行う／行わないと言う組み合わせによって以下の四種類を考えた．

COS：コサイン類似度を閾値とした HAZOP スコアの平均

2COS：関連するイベント（隣接する UML コンポーネント）のタグも考慮したコサイン類似度を閾値とした HAZOP スコアの平均

COS+MDP：上記の COS を各イベントの報酬とし，それに対してリンクを考慮した MDP アルゴリズムを適用したスコア

2COS+MDP：上記の 2COS を各イベントの報酬とし，それに対してリンクを考慮した MDP アルゴリズムを適用したスコア

SD0 の COS による評価では，380 イベント中，30 個が重要度 5 として上位にランク付けられた．その中でも，例えば「ポジション表示出力判定」などシフトのポジション入力判定に関するシーケンス上のメッセージが数多く上位にランクされた．また，全体 380 イベント中，スコアが 0 より大きいものは 195 イベントであった．2COS の結果は COS の結果と類似していたが，「ポジション入力 1 ミドルウェア」のような大きなまとまりを持つコンポーネントが COS の場合よりも上位にランク付けされる結果となっていた．また，COS よりもより安全側の評価であり，380 イベント中重要度が 0 よりも大きいイベント数は 279 個であった．MDP を考慮した COS+MDP，2COS+MDP は同じ報酬でもリンクするコンポーネントの報酬も考慮しているため，より高い重要度が密集する部分のコンポーネントが高くなる傾向にある．そのため，上位のものがより絞り込まれるような結果となった．実際，SD0 の結果において，COS+MDP で最も高いスコアのコンポーネントは，「ポジション入力 2 確定情報」，「CAN1 送信エラー判定」，「デジタル出力通知」，「ポジション表示出力通知」の四つに絞り込まれ，2COS+MDP で最も高いスコアを示したコンポーネントは 5 つとなった．また，MDP では重要度の密集度合いを考慮しているため，大きなまとまりを持つ機能が重要視される傾向にあり，2COS+MDP ではより大きなまとまりを持つコンポーネント「ポジション入力 1 ミドルウェア」などの評価が高くなっていた．SD1 の結果についても，SD0 と同様であり，シーケンスによる違いはあまり見られなかった．SD0 と SD1 は同じ制御目的のシーケンスであるが，一部のロジックが異なるだけである．そのため，コンポーネントのリンクを考慮する 2COS および MDP による評価で SD0 と SD1 の違いが多少みられたが，大きな違いは見受けられなかった．今回提案した手法ではそのような細かいロジックの違いで結果が大きく影響されないことがわかった．安全性分析の手間を低減させるという観点から結果を検証すると，2COS において安全性分析者が重要と考えているコンポーネントの重要度が 0 にならな



い、つまり、100 程度のイベントを分析対象から外すことができているという結果が得られた。

一方、スコアリングのもととして 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、全体的な類似度が低下する結果となった。そのため、閾値を 0.8 と設定した場合、類似するイベントが COS, 2COS いずれも存在しないという結果となった。次に、閾値を 0.6 に下げると、ある特定のイベントがシーケンス上のほとんどのイベントと類似するという結果になった。具体的に、SD0 の場合、全 380 イベント中 361 個のイベントが COS において最も高い重要度 0.25 を示す結果となった。これは本来の目的である、リスク分析を行うコンポーネントを絞り込むという役割を果たしていないため、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性が生じることがわかった。

## 2) 実証分析 (B 社)

### a. 利用データ

利用する設計データは、空調制御システム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際のイベント数は 24 イベントであった。HAZOP2 は ASIL によって予め重要度が与えられていることに注意する。

### b. 分析結果と考察

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 と低めに設定した。

COS: コサイン類似度を閾値とした HAZOP スコアの平均

2COS: 関連するイベント (隣接する UML コンポーネント) のタグも考慮したコサイン類似度を閾値とした HAZOP スコアの平均

COS+MDP: 上記の COS を各イベントの報酬とし、それに対してリンクを考慮した MDP アルゴリズムを適用したスコア

2COS+MDP: 上記の 2COS を各イベントの報酬とし、それに対してリンクを考慮した MDP アルゴリズムを適用したスコア

COS による評価では、24 イベント中、「EEPROM」、「指定ブロック書込」、「EEPROM ミラーバッファ」の三つが重要度 0.25 として上位にランク付けされた。またそれ以外のイベントの重要度は 0 となった。COS+MDP でも「EEPROM」、「指定ブロック書込」、「EEPROM ミラーバッファ」は自身が報酬 (COS による重要度) を有しているためにそれらが上位になるが、「EEPROM」、「指定ブロック書込」、「EEPROM ミラーバッファ」の報酬からその周辺の「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」も比較的高い重要度になった。2COS, 2COS+MDP については、先の A 社の結

果と同様に、より大きな単位の機能が上位に位置づけられる結果となった。実際の開発者からのヒアリングでは、「Backup Manager」などが重要であるという指摘があったが、それは2COS+MDPによる評価では24コンポーネント中14位に位置づけられており、精度があまり高くなかった。この原因として、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考えられる。

### 3.5.3 発生した問題および今後の展望

#### (1) 発生した問題

A社およびB社とのヒアリングでは、ツールの拡張可能性も含めて様々な意見が挙げられた。まず、ツールそのものの問題点として、以下の点が挙げられた。

##### ① HAZOP 分析結果の入力に対する労力

本ツールではHAZOP分析結果を特定のXML形式による書式に変更する必要がある。しかしながら、これらの分析には特定の書式が存在せず、分析者あるいは設計者によって書式が微妙に異なるため、自動的にデータベースに蓄える機能を実現するのが難しいという点が指摘された。この問題を本質的に解決するためには、分析に対応する設計書へのポイントや安全要求仕様へのポイントなどの情報も含めて、標準的な書式を規定する必要がある。実際、そのような標準化の整備や導入は企業にとって容易でないことも事実である。そこで、将来的な可能性としては、任意のフォーマットで文脈を解析して、ある程度自動的に結果を整理する機能を実現することが重要になると考えられる。

##### ② 重要度スコアリングの精度

重要度スコアリングの精度について、A社の実証分析では、重要度をつけることによってある程度のフィルタリング（分析しなくても良い部分を除外することなど）が可能になった。しかしながら、380イベント中の100イベント程度を除外できたに過ぎず、精度については実用レベルとしてまだ十分とは言えない。本研究でとられたアプローチでは単語による類似性を見ているので、HAZOP分析の相互関係や設計で利用する単語を整備することで精度が向上するものと考えられる。また、専門用語や特定の企業だけで利用する略語などに関するベクトル化が適切にできれば、さらなる精度が期待できると考えられる。

##### ③ 出力形式

出力形式については、現在はリスト形式で最低限のシーケンス上の座標情報などを表示しているが、より利用価値の高いツールにするためには視覚的にもっとわかりやすい表示の工夫が必要である。例えば、シーケンス上の危険な部分をヒートマップのように表示するなどの工夫が考えられる。

#### (2) 今後の展望

A社およびB社とのヒアリングの中で、本研究で開発したツール（分析手法）の将来的な可能性について一定の評価が得られた。一方、類似検索だけの機能についても様々な利用上のアイデアが議論された。例えば、過去の設計情報やバグ情報を整理することで、経験の少

ない設計者を「設計」の観点からサポートするシステムや、全体の設計方針を決める際に、より関連するワードがどの程度あるかなどを過去の設計事例から調査するシステムの開発などが挙げられた。また、より細かい粒度においてプログラムソースコード単位で同様な技術を適用し、プログラムそのものの安全性分析も評価できる潜在的な可能性があることが指摘された。

## 4 考察

### 4.1 研究による効果や問題点等

本研究では、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解析した。特に、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。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。そこで、本研究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、注意深く辞書の整理を行うことでリスク分析の精度が高まることがわかった。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 分析結果に予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。重要度をつけることによって、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実現できた。しかしながら、リスク分析の精度はまだ十分ではなく、実用に向けては辞書の整理による類似度を高めることで、重要度の評価精度を向上する必要がある。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何らかの方法で算出することも必要である。さらには、実際問題として同一ドメインに対する HAZOP 分析がない場合への対応なども考える必要がある。本研究では、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予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ような可能性も考えられる。

今回開発したツールそのものについては、プロトタイプであるため、入力およ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。特に、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示すことで、重要な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 形式に限定されており、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。これらについては、統合開発環境との融合により解決すべき問題でもある。さらに、word2vec によるデータの管理や、イベント、タグの管理も現状ではあまりスケーラブルな設計になっておらず、多くのデータから分析を行う場合などより現実的な状況においては分散処理などの方式を検討することが必要である。

## 4.2 産業界への展開と今後の研究の進め方

### 4.2.1 研究成果の産業界への展開

今回開発したツールは機能安全に関連した安全性分析が要請される分野（自動車制御など）において、継続的に安全性分析事例をデータ化することで安全性分析の工数を削減するのにある程度有効であると考えられる。例えば、作成した設計（シーケンス図）に対して安全性分析者がHAZOP安全性分析を行う際、ツールで得られた下位数件のイベント（シーケンス図の一部分）を省略することで工数の削減を行うなどの利用方法が考えられる。今後は、IPA-SECにおける報告会、各種外部イベントや学会発表を通じて、これら分野への普及を行う。また、開発したツール（特に検索エンジンとスコアリングアルゴリズム）はフリーウェアとして一般公開することを検討している。さらに、システム開発会社やコンサルタント会社の協力の下で、協力企業を増やしながらツールを継続的に活用し、利用実績を積み上げながら改良することも予定している。

本研究で開発したようなリスク分析ツールを汎用ツールとして実際の開発現場でより効果的に利用するためには、ツールベンダーとの共同研究開発が不可欠であると思われるが、これを実現するためには先に述べたいくつかの技術的課題を克服しなければならない。

### 4.2.2 今後の研究の進め方

研究プロジェクトを通じて得られた成果の全体を俯瞰するサーベイ論文をSEC Journalに投稿し、日本国内において研究成果の紹介やツールの啓蒙を行うための活動を行う。特に、理論的な研究成果ならびに実証研究の成果については、関連の国際会議や学術雑誌に論文として投稿することを考えている。

開発のツールの有効性や問題点を確認しながらブラッシュアップするためには、ツールの使用許諾を企業に無償で行うだけでは不十分であり、ツールに基づいた定性的信頼性／安全性解析の成功実績を積み上げることが必要である。よって、学会やセミナーでの講演だけでなく、企業との共同研究を積極的に行うことで、ツールの性能向上や拡張に繋がるものと考えられる。

### 4.2.3 産業界への要望

産業界への要望としては、まず基礎的な単語などの整理の必要が挙げられる。文書を形態素解析によって分析する手法によって様々なソフトウェア設計を支援することは今後ますます増えるものと予想される。その際、標準的な単語を規定しておくことは非常に重要である。本研究におけるアプローチに限定した場合でも、重要度スコアリングの精度はword2vecに基づいた類似度に依存しており、さらに、word2vecはMeCabによる形態素解析できちんと品詞に分解される必要がある。特に専門的な用語については、情報が整理されていることが少ないため、これらの単語が含まれている文書が正しく解析されていないことがある。そのため、専門的かつ基礎的な単語を整理し、辞書として提供されることが重要であると考えられる。

次に安全性分析については、既定のフォーマットがなく、自動的なデータ収集が難しい点が挙げられる。自動的にデータを収集し、データベース化することは過去の設計や分析事例を検索するという意味においても重要である。また、フォーマットを整理することで、標準

的なリスク分析の手法を定義できるという利点もある。そのため、適切なフォーマットおよびその作成を支援する統合開発環境が重要になる。

## 参考文献

- [1] P.N. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining: Pearson New International Edition, Pearson Education, 2014.
- [2] 岩田 具治, トピックモデル:機械学習プロフェッショナルシリーズ, 講談社, 2015.
- [3] G. Salton, A. Wong and C.S. Yang, A vector space model for automatic indexing, Communications of the ACM, 18, 613-620, 1975.
- [4] T. Mikolov, K. Chen, G. Corrado and J. Dean, Efficient estimation of word representations in vector space, arXiv preprint arXiv: 1301.3781, 2013.
- [5] H. Kashima, K. Tsuda and A. Inokuchi, Marginalized kernels for labeled graphs, Proceedings of the 20th International Conference on Machine Learning, 321-328, 2003.
- [6] 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, 230-237, 2004.
- [7] 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), 210-219, 2006.
- [8] A. J. Dobson, An Introduction to Generalized Linear Models, Second Edition, Chapman and Hall/CRC, London, 2001.
- [9] S. Brin and L. Page, The anatomy of a large-scale hypertextual Web search engine, Computer Networks and ISDN Systems, 30, 107-117, 1998.
- [10] L. Page, S. Brin, R. Motwai and T. Winograd, The PageRank citation ranking: bringing order to the Web, Stanford Digital Library Technologies, Working Paper 1999-0120, 1998.
- [11] M. Puterman, Markov Decision Processes, John Wiley & Sons, New York, 1994.