

2015 年度ソフトウェア工学分野の先導的研究支援事業

「要求定義の高品質化のための要求仕様の整合性の検証知識の

形式知化と一貫性検証支援ツールの開発」

# シナリオの一貫性検証支援ツール ユーザマニュアル

平成 28 年 2 月

学校法人工学院大学

## 目次

1. はじめに.....	3
2. ツールの機能一覧.....	4
3. インストールのための環境.....	4
4. インストール手順.....	4
4.1 MeCab のダウンロードとインストール.....	4
4.2 MeCab の実行確認 .....	5
4.3 MeCab の辞書を一部変更 .....	5
4.4 検証ツールの入出力フォルダの準備 .....	5
4.5 検証ツールを実行 .....	6
5. 設定について.....	6
5.1 設定ファイル.xml .....	6
5.2 env.xml.....	9
5.3 MeCab 辞書の変更についての補足説明.....	10
6. Input フォルダ内のファイルについて.....	10
6.1 入力設計要素定義表.....	10
6.2 NG ワード定義表.....	11
6.3 識別辞書.....	12
6.4 表記ゆれ辞書 .....	12
7. ルール .....	12
8. 使い方のコツ .....	13
9. ツールのカスタマイズ方法 .....	14

## 1. はじめに

本ドキュメントは、2015 年度ソフトウェア工学分野の先導的研究支援事業「要求定義の高品質化のための要求仕様の整合性の検証知識の形式知化と一貫性検証支援ツールの開発」におけるシナリオの一貫性検証支援ツールのユーザマニュアルである。本ツールは、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」の支援に基づき開発したものである。以下、本ツールのインストール手順や設定方法を説明する。

なお、本ツールが検証対象とする案件に関する対象システムについて補足する。

### (1) 新規 or マイグレーション

- ・特に指定はしていない。
- ・新規とマイグレーションの間の差分を検証する使い方が可能である。

### (2) 規模

- ・特に指定はしていない。
- ・小規模のものについては、人手でチェックすることも可能だが、上限は設定していない。
- ・大規模のものは検証レポートを分割して出力することが望ましい。

### (3) 対象領域

- ・基本的な対象領域は、エンタープライズ系システムとしている。
- ・共通の辞書が、アクター、データ、画面、振る舞いなどエンタープライズ系システムの設計要素を対象にしているためである。
- ・検証の対象を「デバイス」などに拡張することは可能である。

## 2. ツールの機能一覧

本ツールは、5つの機能から構成されており、機能の仕様を表1に示す。

表 1 要求仕様の一貫性検証支援ツール 機能一覧

No.	機能名	ID	処理概要
1	定義漏れ検証	F1	検証対象の仕様(シナリオ)および、アクター、データ、画面、振る舞い等の定義表を入力として、シナリオに出現する、アクター、データ、画面、振る舞い等が、それぞれ、対応する定義表に定義されていることを確認し、未定義であれば指摘する。
2	用語不一致検証	F2	検証対象の仕様(シナリオ)および、アクター、データ、画面、振る舞い等の定義表を入力として、シナリオに出現する、アクター、データ、画面、振る舞い等が、それぞれ、対応する定義表と別の用語表現(表記ゆれ)で記述されている場合に、その表記ゆれを指摘する。
3	用語定義完全一致検証	F3	検証対象の仕様(シナリオ)および、アクター、データ、画面、振る舞い等の定義表を入力として、シナリオに出現する、アクター、データ、画面、振る舞い等が、それぞれ、対応する定義表に定義されていること、かつ、アクター、データ、画面、振る舞い等の定義表に定義された各要素が、シナリオ中に1回以上出現していることを確認し、未定義または出現しない場合に指摘する。
4	NGワード検証	F4	検証対象の仕様(シナリオ)および、NGワード定義表を入力として、NGワードの定義表に定義された用語がシナリオ中に出現していれば、それを指摘する。
5	シナリオ自動補正	F5	検証対象の仕様(シナリオ)および、NGワード定義表を入力として、NGワードの定義表に定義された用語がシナリオ中に出現していれば、おなじくNGワード定義表に定義された置換候補用語でシナリオを置換し、改善シナリオを生成する。

## 3. インストールのための環境

本ツールは、OS Windows 7/8.1/10<sup>1</sup>の環境下で動作可能である。

アカウント名は半角英数字で構成する。

## 4. インストール手順

インストール手順は 4.1～4.5 で説明する。

### 4.1 MeCab のダウンロードとインストール

(1) バイナリパッケージをダウンロードする。

以下のサイトを開く。

<http://taku910.github.io/mecab/#download>

※サイト内を「Binary package for MS-Windows」で検索し、そのすぐ下の「mecab-0.996.exe」をダウンロードする。

(2) ダウンロードしたファイルを実行する。

<sup>1</sup> Windows 7、Windows 8.1、Windows 10 は、米国 Microsoft Corporation の米国およびその他の国における登録商標です。

※以降は、セットアップウィザードに従ってインストールする。

ただし、下記の問いには、以下の通り答える。

- ・辞書の文字コードは「UTF-8」を選択する。
- ・インストール先はデフォルトの「C:\Program Files (x86)\MeCab」とする。

## 4.2 MeCab の実行確認

- (1) 形態素解析したい文章をテキストファイルにして任意の場所に保存する。

※今回は、「C:\Temp\in.txt」に保存したと想定する。

※当ファイルの文字コードは「UTF-8」とする。

- (2) コマンドプロンプトを実行し、下記のコマンドを実行する。

`"C:\Program Files (x86)\MeCab\bin\mecab.exe" C:\Temp\in.txt -o C:\Temp\out.txt`

※「C:\Temp\out.txt」は解析結果を出力するファイルパスである。

- (3) 解析が正しく実行された場合、「C:\Temp\out.txt」というファイルができる。

## 4.3 MeCab の辞書を一部変更

- (1) アクセス許可設定されていない場合、辞書を変更する前に、アクセス許可設定をする必要がある。

(1-1) C:\Program Files (x86)\MeCab\dic\ipadicにある「unk.def」ファイルを右クリックする。

(1-2) プロパティを選択→セキュリティタブ→編集ボタンをクリックする。

(1-3) Users を選択し、フルコントロールの許可の欄にチェックを入れる。

(1-4) すべての許可の欄にチェックがついたのを確認し、適用ボタン→OK をクリックする。

- (2) 「C:\Program Files (x86)\MeCab\dic\ipadic\unk.def」を開く。

- (3) ファイルの以下の行を変更する。

変更前：SYMBOL,1283,1283,17585,名詞,サ変接続,\*,\*,\*,\*,\*

変更後：SYMBOL,1283,1283,17585,記号,一般,\*,\*,\*,\*,\*

- (4) 辞書をコンパイルする。

「Recompile UTF-8」を実行する。

## 4.4 検証ツールの入出力フォルダの準備

- (1) C ドライブの直下に「VerificationTool」を作成する。

- (2) C:\VerificationTool の直下に実行ファイルをダウンロードする。

※「VerificationTool\bin」フォルダを「C:\VerificationTool」フォルダにコピーする。

- (3) 必要に応じて「C:\VerificationTool\bin\検証ツール実行.bat」のショートカットをデスクトップに作成する。

- (4) 「C:\VerificationTool\bin\env.xml」の設定内容に基づき、次の手順で input と output の環境を作成する。以下では、デフォルトの env.xml の設定内容に従って説明する。

(4-1) C:\Temp の直下に「input」フォルダを作成する。

(4-2) (4-1) で作成した input フォルダ直下に「scenario」フォルダを作成する。

(4-3) (4-1) で作成した input フォルダ直下に以下のファイルを置く。

- ・ 設定ファイル（デフォルト：設定ファイル.xml）
- ・ 入力設計要素定義表（デフォルト：入力設計要素定義表.csv）
- ・ NG ワード定義表（デフォルト：NG ワード定義表.csv）
- ・ 識別辞書（デフォルト：識別辞書 1.csv）
- ・ 表記ゆれ辞書（デフォルト：表記ゆれ辞書 1.csv）

(4-4) C:\¥Temp の直下に「output」フォルダを作成する。

(4-5) (4-4) で作成した output フォルダ直下に「scenario」フォルダを作成する。

## 4.5 検証ツールを実行

(1) 4.4 の (4-2) で作成した「scenario」フォルダの中に検証したい入力シナリオのテキストファイルを置く。

※Word,Excel,pdf 等で記述されたシナリオは、あらかじめ、テキストファイル形式に変換しておく。

※テキストファイルの文字コードは「ANSI」で保存する。

(2) 4.4 の (3) で作成したショートカットをダブルクリックする。

output フォルダに「検証レポート」が出力される。

output フォルダの scenario フォルダには、NG ワードを置き換えたシナリオが自動生成される。

（設定ファイルで、シナリオ自動生成を選択した場合に限る。デフォルトの設定ファイルではシナリオを自動生成する設定となっている。）

## 5. 設定について

### 5.1 設定ファイル.xml

設定ファイルとは、利用者が検証条件である検証対象（アクター、データ、画面、振る舞い）、検証（定義漏れ検証、表記ゆれ検証、完全一致検証）の実施有無、NG ワード検証およびシナリオ自動補正の実施有無、品詞別用語ログの出力有無、利用者指定の辞書および検証ルールを指定するための xml ファイルである。

デフォルトで指定されている「設定ファイル.xml」を用いて説明する。図 1 と図 2 は「設定ファイル.xml」である。

```

<?xml version="1.0" encoding="utf-8"?>
<Configuration>
  <アクター>
    <定義漏れ検証>1</定義漏れ検証>
    <表記ゆれ検証>1</表記ゆれ検証>
    <完全一致検証>1</完全一致検証>
  </アクター>
  <データ>
    <定義漏れ検証>1</定義漏れ検証>
    <表記ゆれ検証>1</表記ゆれ検証>
    <完全一致検証>1</完全一致検証>
  </データ>
  <画面>
    <定義漏れ検証>1</定義漏れ検証>
    <表記ゆれ検証>1</表記ゆれ検証>
    <完全一致検証>1</完全一致検証>
  </画面>
  <振る舞い>
    <定義漏れ検証>1</定義漏れ検証>
    <表記ゆれ検証>1</表記ゆれ検証>
    <完全一致検証>1</完全一致検証>
  </振る舞い>
  <NG ワード検証>1</NG ワード検証>
  <シナリオ自動補正>1</シナリオ自動補正>
  <品詞別用語ログ出力>1</品詞別用語ログ出力>
  <識別辞書>
    <アクター>
      <辞書>識別辞書 1 .csv</辞書>
    </アクター>
    <データ>
      <辞書>識別辞書 1 .csv</辞書>
    </データ>
    <画面>
      <辞書>識別辞書 1 .csv</辞書>
    </画面>
    <振る舞い>
      <辞書>識別辞書 1 .csv</辞書>
    </振る舞い>
  </識別辞書>

```

図 1 設定ファイル.xml (1/2)

```

<検出ルール>
  <アクター>
    <ルール>actor-rules-01.rb</ルール>
  </アクター>
  <データ>
    <ルール>data-rules-01.rb</ルール>
  </データ>
  <画面>
    <ルール>screen-rules-01.rb</ルール>
  </画面>
  <振る舞い>
    <ルール>action-rules-01.rb</ルール>
  </振る舞い>
</検出ルール>
<表記ゆれ検出辞書>
  <アクター>
    <辞書>表記ゆれ辞書 1 .csv</辞書>
  </アクター>
  <データ>
    <辞書>表記ゆれ辞書 1 .csv</辞書>
  </データ>
  <画面>
    <辞書>表記ゆれ辞書 1 .csv</辞書>
  </画面>
  <振る舞い>
    <辞書>表記ゆれ辞書 1 .csv</辞書>
  </振る舞い>
</表記ゆれ検出辞書>
<設計要素検出優先順位>
  <アクター>3</アクター>
  <データ>1</データ>
  <画面>2</画面>
  <振る舞い>4</振る舞い>
</設計要素検出優先順位>
</Configuration>

```

図 2 設定ファイル.xml (2/2)

図 1 の上部は、検証の対象（アクター、データ、画面、振る舞い）、検証の種類（定義漏れ検証、表記ゆれ検証、完全一致検証）、NG ワード検証、シナリオ自動補正の実施有無、品詞別用語ログの出力有無が記述されている。それぞれのタグ（<定義漏れ検証>、<表記ゆれ検証>、<完全一致検証>、<NG ワード検証>、<シナリオ自動補正>、<品詞別用語ログ出力>）に挟まれた“1”は、検証を実施することを示している。（実施しない場合は、“0”を記述する）

図 1 の<識別辞書>から</識別辞書>タグで挟まれた部分は、アクター、データ、画面、振る舞いそれぞれに使用する識別辞書を指定する記述で、デフォルトで「識別辞書 1.csv」が指定されている。

図 2 の<検出ルール>から</検出ルール>タグで挟まれた部分は、アクター、データ、画面、振る舞いそれぞれに使用するルールを指定する記述で、アクターの検出ルールは、デフォルトで「actor-rules-01.rb」が指定されている。

図 2 の<表記ゆれ検出辞書>から</表記ゆれ検出辞書>タグで挟まれた部分は、アクター、データ、画面、振る舞いそれぞれに使用する表記ゆれ辞書を指定する記述で、デフォルトで「表記ゆれ辞書 1.csv」が指定されている。



図 2 の<設計要素検出優先順位>から</設計要素検出優先順位>タグで挟まれた部分は、アクター、データ、画面、振る舞いの設計要素を抽出するときの優先順位を指定する記述である。

## 5.2 env.xml

env.xml は、利用者が検証の入出力ファイル名、パス、つなぎ文字や接尾語の扱いを指定するためのファイルである。デフォルトでは、図 3 に示す内容が C:¥VerificationTool¥bin の中に定義されている。図 3 の env.xml 内のタグとその意味を表 2 に示す。

```
<?xml version="1.0" encoding="utf-8"?>
<data>
  <input>
    <dir_path>C:¥Temp¥input</dir_path>
    <scenario_dir_nm>scenario</scenario_dir_nm>
    <config_file_nm>設定ファイル.xml</config_file_nm>
    <definition_table_file_nm>入力設計要素定義
表.csv</definition_table_file_nm>
    <ng_word_file_nm>NG ワード定義表.csv</ng_word_file_nm>
  </input>
  <output>
    <dir_path>C:¥Temp¥output</dir_path>
    <report_file_nm>検証レポート.csv</report_file_nm>
    <scenario_dir_nm>scenario</scenario_dir_nm>
  </output>
  <connection_char>
    <char>の</char>
    <char>・</char>
    <char>/</char>
    <char></char>
  </connection_char>
  <suffix>
    <char>毎</char>
    <char>等</char>
    <char>など</char>
    <char>等々</char>
    <char>etc.</char>
    <char>上</char>
    <char>中</char>
    <char>下</char>
    <char>内</char>
    <char>外</char>
  </suffix>
</data>
```

図 3 env.xml

表 2 env.xml 内のタグとその意味

項番	タグ	タグの意味
1	<input>	ツールに入力する検証対象のフォルダ名や配置先、設定ファイル・入力設計要素定義表・NG ワード定義表の名称を指定する。
2	<dir_path>	ツールに入力するフォルダのパスを指定する。
3	<scenario_dir_nm>	入力シナリオを入れるフォルダ名を指定する。
4	<config_file_nm>	検証対象等を指定する設定ファイルの名称を指定する。

5	<definition_table_file_nm>	入力設計要素定義表の名称を指定する。
6	<ng_word_file_nm>	NG ワード定義表の名称を指定する。
7	<output>	検証結果の出力先となるフォルダ名や配置先、検証レポートの名称、代替シナリオを入れるフォルダ名を指定する。
8	<dir_path>	出力先となるフォルダのパスを指定する。
9	<report_file_nm>	検証レポートの名称を指定する。
10	<scenario_dir_nm>	代替シナリオを入れるフォルダ名を指定する。
11	<connection_char>	「の」、「・」、「／」、「/」等のつなぎ文字を指定する。つなぎ文字で連結された設計要素用語も 1 つの用語として抽出するために必要である。つなぎ文字は、追加または削除できる。
12	<suffix>	「毎」、「等」、「など」、「等々」等の接尾語を指定する。接尾語が付いた設計要素用語も 1 つの用語として抽出するために必要である。接尾語は、追加または削除できる。

### 5.3 MeCab 辞書の変更についての補足説明

#### (1) 変更内容

「C:\Program Files (x86)\MeCab\dic\ipadic\unk.def」を以下のとおり変更する。

変更前：SYMBOL,1283,1283,17585,名詞,サ変接続,\*,\*,\*,\*,\*

変更後：SYMBOL,1283,1283,17585,記号,一般,\*,\*,\*,\*,\*

#### (2) 変更の意図

「C:\Program Files (x86)\MeCab\dic\ipadic\char.def」にて文字コード毎に定義された SYMBOL に相当する語を、「名詞，サ変接続」から「記号，一般」と認識するように変更するものである。

#### (3) MeCab の運用への影響

デフォルトの MeCab では、「C:\Program Files (x86)\MeCab\dic\ipadic\Symbol.csv」において特定の SYMBOL（全角記号）は「記号，一般（または句点、読点、括弧開、括弧閉、アルファベット）」と認識するよう定義され、「Symbol.csv」において定義されていない SYMBOL（半角記号）は、「unk.def」で「名詞，サ変接続」と認識するように定義されている。よって、デフォルトの MeCab では全角記号は「記号，一般（または句点、読点、括弧開、括弧閉、アルファベット）」、半角記号は「名詞，サ変接続」と出力される。しかし、「unk.def」を（2）のように変更することによって、全角記号は「記号，一般（または句点、読点、括弧開、括弧閉、アルファベット）」、半角記号は「記号，一般」と出力できるようになる。したがって、「Symbol.csv」内で未定義の SYMBOL を新たに「記号，一般（または句点、読点、括弧開、括弧閉、アルファベット）」として定義せずとも、MeCab の運用に問題はない。

## 6. Input フォルダ内のファイルについて

### 6.1 入力設計要素定義表

入力設計要素定義表は、csv ファイルで記述する。シナリオに出現するアクター、データ、画面、振る舞いを定義することで、表記ゆれや定義漏れの検出ができる。ツールのサンプルデータとして提供している入力設計要素定義表.csv を表 3 に示す。利用者は入力設計要素定義表に用語を追加または削

除できる。1 レコードは、ラベル、No. (連番)、要素定義で構成する。ラベルは、Actor、Data、Screen、Action の 4 つを定義している。それぞれ、仕様書中にて定義されている、アクター、データ、画面、振舞いに対応する。

表 3 入力設計要素定義表.csv

Actor	1 顧客
Actor	2 会員
Actor	3 店員
Actor	4 システム管理者
Actor	5 販売管理システム
Data	1 商品情報
Data	2 顧客情報
Data	3 ポイントカード情報
Screen	1 ポイントカードログイン画面
Screen	2 ポイントカード実績入力画面
Screen	3 ポイントカード実績確認画面
Action	1 登録する
Action	2 更新する
Action	3 検索する
Action	4 出力する

## 6.2 NG ワード定義表

NG ワード定義表は、シナリオ中に使用すべきではない用語を定義した表である。デフォルトで指定されている NG ワード定義表.csv の一部を表 4 に示す。NG ワード定義表.csv は、文献と有識者のインタビューで得た知見を基に作成した。利用者は NG ワード定義表に用語を追加または削除できる。1 レコードは、No. (連番)、NG 用語、置換候補用語、備考で構成する。

表 4 NG ワード定義表.csv

No.	NG用語	置換候補用語	備考
1	すべて		
2	全て		
3	あらゆる		
4	既存システムと同様		
5	既設システムと同様		
6	ユーザー		
7	利用者		
8	受け入れ可能な		受け入れられるものは何か、そしてシステムはどうやってそれを判断できるかを定義する (Wiegers 2013)
9	適切な		受け入れられるものは何か、そしてシステムはどうやってそれを判断できるかを定義する (Wiegers 2013)
10	そして/または		「そして」なのか、「または」なのか、それともどんな組み合わせでもいいのかを指定して、読者が推測しなくてもいいようにする (Wiegers 2013)
11	実行できる程度に		実行できるとは何を指すのか、という決定を開発者に委ねない。TBDとし、解決期限の日付を設定する (Wiegers 2013)
12	約		
13	だいたい		
14	最新版		
15	～しないわけではない		
16	少なくとも		受け入れ可能値の最大値と最小値を指定する (Wiegers 2013)
17	最低でも		受け入れ可能値の最大値と最小値を指定する (Wiegers 2013)
18	せいぜい		受け入れ可能値の最大値と最小値を指定する (Wiegers 2013)
19	上回らない		受け入れ可能値の最大値と最小値を指定する (Wiegers 2013)
20	最高の		どの程度の達成度が望まれるかを指定して、最低限受け入れ可能な達成度を指定する (Wiegers 2013)
21	最善の		どの程度の達成度が望まれるかを指定して、最低限受け入れ可能な達成度を指定する (Wiegers 2013)
22	最大の		どの程度の達成度が望まれるかを指定して、最低限受け入れ可能な達成度を指定する (Wiegers 2013)
23	～の間		両端の値が範囲に含まれるかどうか定義する (Wiegers 2013)

### 6.3 識別辞書

識別辞書は、ツールがシナリオに出現する用語をアクター、データ、画面、振る舞いに識別するための辞書である。デフォルトで指定されている識別辞書 1.csv を表 5 に示す。利用者は識別辞書に用語を追加または削除できる。1 レコードは、ラベル、No. (連番)、識別用語で構成する。

表 5 識別辞書 1.csv

ラベル	No.	識別用語
Actor	1	者
Actor	2	部
Actor	3	部門
Actor	4	会社
Actor	5	局
Actor	6	課
Actor	7	グループ
Screen	1	ページ
Screen	5	ウェブサイト
Screen	6	ホームページ
Screen	7	メッセージ
Action	1	する
Action	2	実施
Action	3	実行
Action	4	管理

### 6.4 表記ゆれ辞書

表記ゆれ辞書は、シナリオ中に出現するアクター、データ、画面、振る舞いの表記ゆれだと考えられる用語を定義する。実際に表記ゆれ辞書で定義した表記ゆれの利用語がシナリオ中に出現する場合表記ゆれしている用語と修正候補を検証レポートに出力する。例えば、“ユーザ”であったら、表記ゆれとして、“利用者”や“ユーザー”が考えられる。表 6 のように利用者が表記ゆれ辞書のファイルに記述することで、用語不一致検証が行える。1 レコードは、ラベル、No. (連番)、設計要素、表記ゆれ候補で構成する。

表 6 表記ゆれ辞書 1.csv

ラベル	No.	設計要素	表記ゆれ候補
Actor	1	ユーザ	利用者,ユーザー

## 7. ルール

本ツールにデフォルトで定義している検証ルール、識別ルールの一覧を付録に示す。以下にデフォルトの定義に従って検証した場合に得られる結果の解釈の方法、検証にあたっての留意点についてまとめる。

#### (1) 数字と名詞の結合

本ツールは、数字の後に接尾やサ変接続の名詞があると数字と名詞が結合した形で抽出する。例えば、“100 人”は数字と名詞を結合し、“100 人”の形で抽出する。

#### (2) “,” (全角のカンマ) で名詞を連結するときの注意点

シナリオ中で「名詞、接尾」となる用語を“,”（全角のカンマ）を用いて連続して記述した場合、アクターとして連結して認識されることがある。例えば、“全社、局、課”は、“局”と“課”が「名詞、接尾」に相当するので **MeCab** によって連結して“全社、局、課”の形で抽出され、本ツールはアクターとして認識する。これは、**Mecab** において“,”（全角のカンマ）が「名詞、数」と抽出されてしまうことによる。シナリオ中で名詞を“,”（全角のカンマ）を用いて並べる際には注意が必要である。“,”（全角のカンマ）“、”や“,”（半角のカンマ）で代替することで問題なく本ツールを運用することができる。

### (3) 「修飾語が付与された名詞」の抽出

「修飾語が付与された名詞」は、本ツールでは名詞のみを抽出する。例えば、“設定したページ”や“登録されているページ”は“ページ”のみを抽出する。

### (4) 「目的語＋動詞句」の扱い方

本ツールは、目的語と動詞句を一緒に振る舞いとして抽出しない。例えば、“注文を登録する”からは“登録する”を振る舞いとして抽出する。

### (5) 品詞が妥当に認識されない例外

**MeCab** の仕様で品詞が妥当に認識されないケースがある。例えば、本ツールは“障がいのある人”をアクターとして認識しない。これは、**MeCab** において“障”が動詞の自立、“がい”が名詞の非自立と認識され、“障”と“がい”が別々に抽出されることによる。尚、本ツールは“〇〇のある人”の形で抽出しないため、“障がい”を“障害”や“障碍”にした場合でも“人”のみをアクターとして抽出する。

### (6) “(” “)”を含む用語の抽出

本ツールでは、“(” “)”を含む複合語全体を名詞として抽出しない。例えば、“担当（作成）部署”の場合、“担当（作成）”と“部署”に分けて抽出する。尚、“（作成）”の後ろの“部署”を辞書で定義すれば、“担当（作成）部署”の形で抽出することができる。

### (7) “できる”の抽出

本ツールは、「サ変接続名詞＋できる」の形であれば、“〇〇できる”の形で抽出する。例えば、“注文できる”は“注文”が「サ変接続名詞」なので、“注文できる”の形で抽出する。一方で、“切り替えできる”のような「自立動詞＋できる」や“プレビューできる”のような「一般名詞＋できる」の場合は“〇〇できる”の形では抽出しないので注意が必要である。

## 8. 使い方のコツ

要求定義工程での高品質なシナリオの定義のために用いるのではなく、様々なドキュメントに対して本ツールを適用することが想定できる。以下、例を示す。

- ・ 操作仕様書や製品取扱い説明書
- ・ 要求定義書と基本設計書間の比較

- ・プロダクトライン型開発のモデル間の比較
- ・業務マニュアルからアクター用語抽出

## 9. ツールのカスタマイズ方法

設定ファイル名、検証レポートファイル名、input や output フォルダのパスは「bin¥env.xml」ファイルにおいて変更することが可能である。また、同ファイルに内において、設計要素名を識別する際に、名詞と名詞を連結する役割として認識すべき助詞「の」や記号「・」の定義や、名詞の後に接続して同一の名詞として認識すべき接尾語「等」や「毎」の定義を記述することができる。

### [参考文献]

- [1] MeCab: Yet Another Part-of-Speech and Morphological Analyzer,  
<http://taku910.github.io/mecab/#download>
- [2] Ruby: <https://www.ruby-lang.org/ja/about/>  
(以下よりインストール実施：<http://rubyinstaller.org/downloads/>)

### [商標表示]

- ・ Windows® operating system, Microsoft Word®, Microsoft Excel®, Microsoft PowerPoint®は、米国 Microsoft Corporation の米国およびその他の国における登録商標または商標です。
- ・ Adobe PDF は、Adobe Systems Incorporated（アドビシステムズ社）の米国ならびにその他の国における商標または登録商標です。

---

### [発行履歴]

2016 年 2 月 12 日 初版発行

以上