保証ケース統合作成支援ツール

操作マニュアル



　　　　　　　　　　　　　　　　　　株式会社　チェンジビジョン

目次

[操作マニュアル 3](#_Toc437003942)

[メニュー 3](#_Toc437003943)

[プロジェクトに関する操作 3](#_Toc437003944)

[プロジェクトを新規作成 3](#_Toc437003945)

[プロジェクトを開く 3](#_Toc437003946)

[プロジェクトを保存 3](#_Toc437003947)

[プロジェクトを名前を付けて保存 4](#_Toc437003948)

[分解に関する操作 4](#_Toc437003949)

[アーキテクチャ分解 4](#_Toc437003950)

[品質特性分解 4](#_Toc437003951)

[障害リスク分解 4](#_Toc437003952)

[編集に関する操作 5](#_Toc437003953)

[重みを編集する 5](#_Toc437003954)

[達成度を編集する 5](#_Toc437003955)

[証拠を編集する 5](#_Toc437003956)

[リスク定義の更新 5](#_Toc437003957)

[要素の削除 6](#_Toc437003958)

[XMI出力 6](#_Toc437003959)

[定義ファイル 8](#_Toc437003960)

[構成定義 8](#_Toc437003961)

[品質特性定義 10](#_Toc437003962)

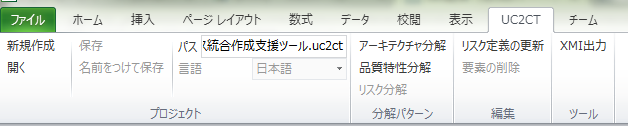
[障害リスク定義 11](#_Toc437003963)

[インストール手順 13](#_Toc437003964)

# 操作マニュアル

## メニュー

保証ケース作成支援ツールの操作は、UC2CTリボンメニューから行う。



## プロジェクトに関する操作

### プロジェクトを新規作成

　　プロジェクトを新規作成するためには、新規作成ボタンを押下する。

　新規作成時、構成定義、品質特性定義、障害リスク定義ファイルが格納されたフォルダの指定、および、プロジェクトで使用する言語の指定を促される。

　　プロジェクト作成と同時に、定義情報及び言語設定に従って、最上位主張が作成さる。

　　作成される主張の内容は、

　　日本語：「{対象モデル名}は、{品質特性名}要件を満たす。」

　　英語：　「{対象モデル名} satisfies {品質特性名} requirement.」.

※{ }は、定義情報から適宜に取得することを示す。以下同様。

### プロジェクトを開く

　　既存の保証ケースプロジェクトを開くためには、開くボタンを押下する。

　　パスに、プロジェクトファイルのパスが表示される。

### プロジェクトを保存

　　編集中の保証ケースプロジェクトを保存するためには、保存ボタンを押下する。

　　プロジェクトファイルの拡張子は.uc2ctとする。

### プロジェクトを名前を付けて保存

　ファイル名を指定して編集中の保証ケースプロジェクトを保存するためには、名前をつけて保存ボタンを押下する。

## 分解に関する操作

### アーキテクチャ分解

　本機能は、最上位主張のみ存在、あるいは、品質特性分解された状態で実行可能となる。

　主張を成果物の構成定義に従って分解するためには、アーキテクチャ分解ボタンを押下する。

　要素・関係層、要素・関係分類層、要素・関係分類実体層の主張が一連で作成される。

　　作成される主張の内容は、

　　　日本語：「{対象モデル名}は、{品質特性名}要件を満たす。」

　　　英語：　「{対象モデル名} satisfies {品質特性名} requirement.」

　　分解時に、構成定義ファイル名を内容とするコンテキストを作成する。

### 品質特性分解

　本機能は、最上位主張のみ存在、あるいは、アーキテクチャ分解された状態で実行可能となる。

　　主張を品質特性の定義に従って分解するためには、品質特性分解ボタンを押下する。

　　作成される主張の内容は、

　日本語：「{対象モデル名}は、{品質特性名}要件を満たす。」

　英語：　「{対象モデル名} satisfies {品質特性名} requirement.」

　　分解時に、品質特性ファイル名を内容とするコンテキストを作成する。

### 障害リスク分解

　　本機能は、アーキテクチャ分解かつ品質特性分解された状態で実行可能となる。

　障害リスクの定義に従って分解するためには、対象とする主張を選択(複数選択可)し、リスク分解ボタンを押下する。

　対象とするリスクの選択を促されるので、選択する。（複数選択可能）

　　作成される主張の内容は、

　日本語：「{対象モデル名}は、{リスク分類名}に対して{品質特性名}要件を満たす。」

　英語：　「{対象モデル名} satisfies {品質特性名} requirement　for {リスク分類名}.」

分解時に、障害リスク定義ファイル名を内容とするコンテキストを作成する。

さらに、リスクに対する主張の子要素としてリスクに対する主張を作成する。

　　作成される主張の内容は、

　日本語：「{対象モデル名}は、{リスク名}に対して{品質特性名}要件を満たす。」

　英語：　「{対象モデル名} satisfies {品質特性名} requirement for {リスク名}.」

また、あわせて、リスクに対する証拠が作成される。

証拠の内容は、

　日本語：「{対象モデル名} {リスク名}{品質特性名}確認報告」

　英語：　「{対象モデル名} {リスク名}{品質特性名} Confirmation Document」

## 編集に関する操作

### 重みを編集する

　　主張に対する重みを編集するには、表の重みのセルを直接編集する。

　　編集すると、上位の主張の達成度を再計算する。

　　各主張の達成度達成度は、下位の主張の達成度×重みの和により求める。

### 達成度を編集する

　　証拠の達成度を編集するには、達成度のセルを直接編集する。

　　編集すると、上位の主張の達成度を再計算する。

　　各主張の達成度達成度は、下位の主張の達成度×重みの和により求める。

### 証拠を編集する

　　証拠の内容を編集するには、証拠のセルを直接編集する。

### リスク定義の更新

　　リスク定義の変更を反映するには、リスク定義更新ボタンを押下する。

　これにより、リスク定義の追加、変更、削除がプロジェクトに反映される。削除されたリスク定義あるいは追加、削除された逸脱に基づいて作成された主張が存在する場合は、当該主張およびそれに対応する証拠が削除される。

### 要素の削除

　選択(複数選択可)した主張を削除するには、リスクに関する主張に対して要素の削除ボタンを押下する。

削除主張に対する証拠も合わせて削除する。

## XMI出力

　保証ケースをSACM(Structured Assurance Case Metamodel) version1.0のスキーマ定義に従ったXMI形式で出力する。

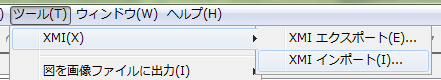
##### astah\* GSNへのインポート手順

　ここで、出力したXMIファイルをastah\* GSNでインポートし、GSNを生成する手順を示す。

1. astah\* GSNを起動する。
2. GSNを作成する。



1. 保証ケース統合作成支援ツールで出力したXMIファイルをインポートする。

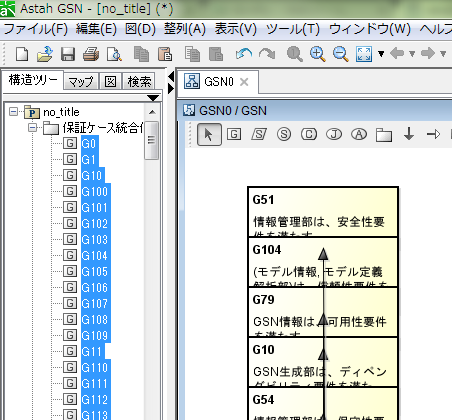


これにより、保証ケースの議論要素がastah\* GSNに取り込まれる。

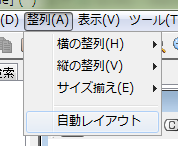
1. 取り込まれた議論要素を選択し、GSNへドラッグ＆ドロップする。

※この時、議論要素以外の要素を選択しないようにすること。例えば、GSN図やモジュールなど。

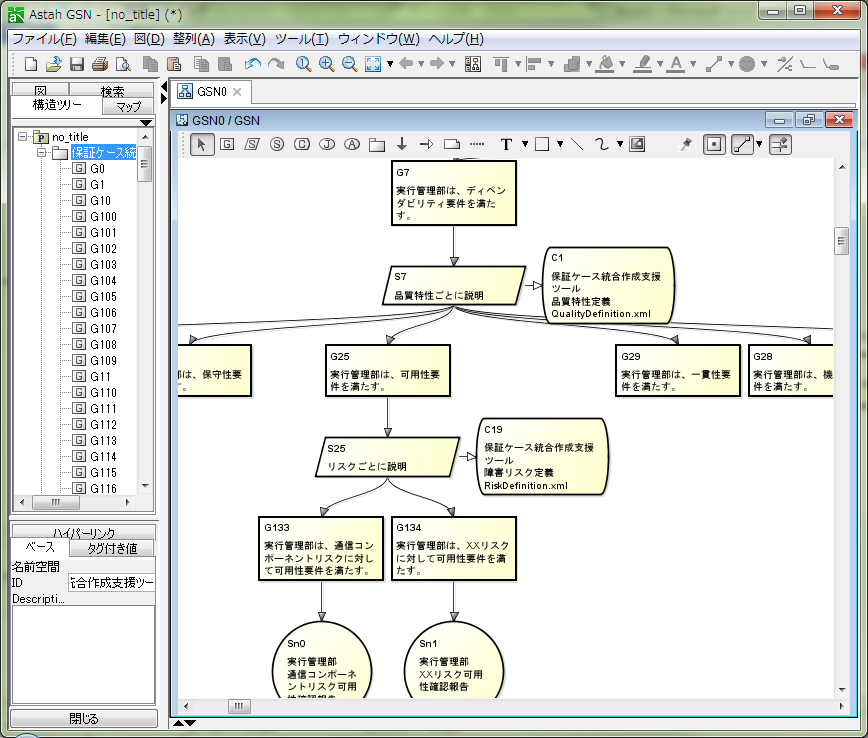
これにより、議論要素がGSNに展開される。



1. GSNに展開された議論要素を自動レイアウトにより整列させる。



1. GSNの完成。



# 定義ファイル

本ツールの入力となる定義ファイルの仕様について記載する。

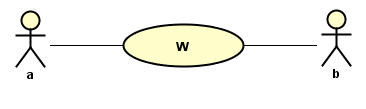
各種定義ファイルは、XML形式とする。

## 構成定義

構成定義ファイル名は、ModelDefinition.xmlとする。

以下に、構成定義の例を示すとともに、定義フォーマットについて説明する。

例では、下記のモデルを定義対象としている。



<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>

<modelDefinition>

　<!-- name属性にはシステム名に相当する名前を定義 -->

<model name="ユースケース図">

　<!-- モデルの型を定義 -->

<types>

<nodes>

<node>Actor</node>

<node>Usease</node>

</nodes>

<relations>

<relation>Actor\_Usecase</relation>

</relations>

</types>

　 <!—- 実体を定義-->

<instances>

<nodes>

<node type="Actor" id="in001">a</node>

<node type="Actor" id="in002">b</node>

<node type="Usecase" id="in003">W</node>

</nodes>

<relations>

<relation type="Actor\_Usecase" id="ir001" source="in001" target="in003" />

<relation type="Actor\_Usecase" id="ir002" source="in003" target="in002" />

</relations>

</instances>

</model>

</modelDefinition>

|  |  |
| --- | --- |
| 要素 | 説明 |
| modelDefinitionタグ | 構成定義のルート要素 |
| modelタグ | モデルの定義群 |
| name属性 | モデル名 |
| typesタグ | 型の定義群 |
| nodesタグ | ノード型の定義群 |
| nodeタグ | ノード型の定義 |
| relationsタグ | 関連型の定義群 |
| relationタグ | 関連の定義 |
| instancesタグ | 実体の定義群 |
| nodesタグ | ノード型の実体の定義群 |
| nodeタグ | ノード型の実体の定義 |
| type属性 | 実体の型の定義 |
| id属性 | 実体のID |
| relatiosタグ | 関連型の実体の定義群 |
| relationタグ | 関連型の実体の定義 |
| type属性 | 実体の型の定義 |
| id属性 | 実体のID |
| source属性 | 関連元となる実体のID |
| target属性 | 関連先となる実体のID |

表 1　構成定義フォーマットの説明

## 品質特性定義

品質特性定義ファイル名は、QualityDefinition.xmlとする。

以下に、品質特性定義の例を示すとともに、定義フォーマットについて説明する。

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>

<qualityDefinition>

　<!-- ルートとなる品質特性についてはroot属性をtrueと定義 -->

<attribute root="true" name="ディペンダビリティ">

　　<!-- struct要素により階層的に定義 -->

<struct>

<attribute-ref>可用性</attribute-ref>

<attribute-ref>信頼性</attribute-ref>

<attribute-ref>安全性</attribute-ref>

</struct>

</attribute>

<attribute name="可用性" />

<attribute name="信頼性" />

<attribute name="安全性">

　　<!-- 品質基準を定義 -->

<criteria name="システム安全管理原則">

<list>

<item>システムの仕様や運用方法を明確に文書化している</item>

<item>システムの仕様や運用方法が当初の方針の通りに機能しているかどうかを定期的に監査している</item>

<item>システムの監査結果をあいまいさのない形で文書化している</item>

<item>システムの監査の結果に問題があった場合は、真摯に対応している</item>

<item>問題対応の記録を文書化し、第三者が検証可能な状況にしている</item>

</list>

</criteria>

</attribute>

</qualityDefinition>

|  |  |
| --- | --- |
| 要素 | 説明 |
| qualityDefinitionタグ | 品質特性定義のルート要素 |
| attributeタグ | 品質特性の定義 |
| root属性 | ルートとなる品質特性の場合trueを指定 |
| name属性 | 品質特性名を定義 |
| structタグ | 品質特性を定義する品質特性の定義群 |
| attribute-refタグ | 品質特性を構成する品質特性への参照を指定 |
| criteria属性 | 品質基準の定義 |
| name属性 | 品質基準名の定義 |
| listタグ | 品質基準項目の定義群 |
| itemタグ | 品質基準項目の定義 |

表 2　構成定義フォーマットの説明

## 障害リスク定義

障害リスク定義ファイル名は、RiskiDefinition.xmlとする。

以下に、障害リスク定義の例を示すとともに、定義フォーマットについて説明する。

<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>

<riskDefinition>

　<!-- リスク分類を定義 -->

<risks>

<risk name="通信コンポーネントリスク">

　　　<!-- struct要素により個別リスクへの参照を定義 -->

<struct>

<deviation-ref>Exception</deviation-ref>

<deviation-ref>Delay</deviation-ref>

</struct>

</risk>

</risks>

　<!--リスクを定義 -->

<deviations>

<deviation name="Exception">

<list>

<item>入力例外</item>

<item>処理例外</item>

<item>出力例外</item>

</list>

</deviation>

<deviation name="Delay">

<list>

<item>入力遅延</item>

<item>処理遅延</item>

<item>出力遅延</item>

</list>

</deviation>

</deviations>

</riskDefinition>

|  |  |
| --- | --- |
| 要素 | 説明 |
| riskDefinitionタグ | リスク定義のルート要素 |
| risksタグ | リスク分類の定義群 |
| name属性 | リスク分類名を定義 |
| structタグ | リスク分類を構成する個別リスクの定義群 |
| deviation-refタグ | リスク分類を構成する個別リスクを指定 |
| deviationsタグ | リスクの定義群 |
| deviationタグ | リスクの定義 |
| name属性 | リスク名の定義 |
| listタグ | 個別リスク項目の定義群 |
| itemタグ | 個別リスク項目の定義 |

表 3　リスク定義フォーマットの説明

# インストール手順

以下に、本ツールのインストール手順について記載する。

1. 別途ご案内のURLまたは提供物からインストーラを取得する。
2. 取得したファイルがZipファイルの場合は、取得したファイルを解凍する。
3. Publish\setup.exeをダブルクリックする。
4. インストール中に以下のエラーが発生した場合



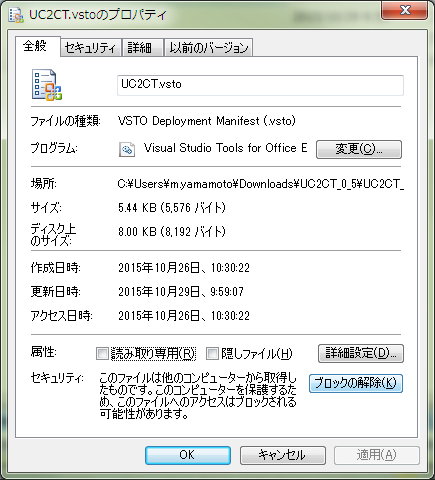
　セキュリティ対策によりインストールできない問題への対処として、ブロックの解除を行う。

　エクスプローラを使用し、以下のファイルについて、ファイルのプロパティ画面でブロックの解除ボタンを押下する。

・\Publish\UC2CT.vsto

・\Publish\Application Files\UC2CT\_0\_5\_0\_0\UC2CT.dll.manifest

・\Publish\Application Files\UC2CT\_0\_5\_0\_0\UC2CT.vsto



1. インストール完了メッセージが表示され、ExcelにUC２CTのメニューが表示されればインストール完了。



