

D-Case Editor の機能拡充に関する開発
機能仕様書

28/MAR/2013

AXE, Inc.

改訂履歴

更新日	版	内容	担当
28/MAR/2013	0.8.9	<ul style="list-style-type: none"> 「3 ノードとリンク」を追加 public/private 属性をモジュールに追加(4.3, 4.4) 図 10 の背景色を橙色に変更 「4.5 モジュール・サブツリーのコピー」を追加 モジュール解除時の処理内容を変更(4.7) 責任属性にアイコン、メールアドレスを追加(4.9) 「Adjust Reference」を「Refresh」に変更(4.10) 日本語訳を追加(4.12) パターンの追加処理内容を変更(5.4.3) パラメータ定義・値の設定を変更(6) 	臼田@AXE
6/MAR/2013	0.8	<ul style="list-style-type: none"> 「4.9 責任属性」を追加 モジュールの同期処理に、責任属性の処理を追加(4.10) 	臼田@AXE
31/JAN/2013	0.7	<ul style="list-style-type: none"> パターンが単一のツリーかどうかのチェックを、メニューの表示段階で行うよう変更(5.4) Desc 属性に設定される旨を追記(6.3) クリップボードにコピーする機能を追加(6.4) 	臼田@AXE
18/JAN/2013	0.6	<ul style="list-style-type: none"> モジュールビューにノード数を追加(4.10) モジュールの参照の同期をとる処理を追加(4.10) 「4.11 ノード一覧の出力」を追加 「4.12 日本語化」を追加 パターンは、モジュールではなくそのまま追加するよう変更(5) パラメータ定義時にデフォルト値を設定(6.2) 「6.4 パラメーター一覧の表示」を追加 	臼田@AXE
28/DEC/2012	0.5	<ul style="list-style-type: none"> main もモジュールとして扱うよう記述を変更(4.2~4.4) モジュールの Refresh 機能は不要のため削除(4.8) 	臼田@AXE
11/OCT/2012	0.4	<ul style="list-style-type: none"> 「4 モジュール」のファイルの格納場所を変更 「4.3.2 既存ノードのモジュール化」で、ルートノードを選択してモジュール化する手順に変更 「5 パターン」を単一のファイル前提として各処理を変更 	臼田@AXE
5/OCT/2012	0.3	<ul style="list-style-type: none"> 「4 モジュール」の構成を変更 「4.2 モジュールの構造」を追加 Module ノードおよび Away Goal ノードに、参照先のパスの表示を追加(4.4) ダイアグラム全体のモジュール内再表示を追加(4.8) 「4.10 モジュールの管理」を追加 「5.3 パターンの構造」を追加 「5.4.3 パターンの追加処理」に詳細の手順を追加 	臼田@AXE
2/OCT/2012	0.2	新規作成(ドラフト)	臼田@AXE

目次

1 はじめに.....	5
1.1 概要.....	5
1.2 用語の定義.....	5
1.3 関連文書.....	5
2 拡張機能概要.....	7
3 ノードとリンク.....	8
3.1 ノード.....	8
3.2 リンク.....	9
4 モジュール.....	10
4.1 概要.....	10
4.2 モジュールの構造.....	10
4.2.1 ファイルの格納場所.....	10
4.2.2 モジュールの参照.....	10
4.3 モジュールの作成.....	11
4.3.1 Module ノードの作成.....	11
4.3.2 既存ノードのモジュール化.....	11
4.4 モジュールの参照.....	12
4.4.1 Module ノードの参照先の指定.....	12
4.4.2 Away Goal.....	13
4.5 モジュール・サブツリーのコピー.....	15
4.5.1 モジュールのコピー.....	15
4.5.2 サブツリーのコピー.....	15
4.6 public/private 属性.....	16
4.7 モジュールの解除.....	16
4.8 モジュールの展開表示.....	17
4.9 責任属性.....	18
4.9.1 責任属性の追加.....	18
4.9.2 責任委譲関係のリンク.....	19
4.9.3 d*表現.....	20
4.10 モジュールの管理.....	22
4.11 ノード一覧の出力.....	23
4.12 日本語化.....	25
4.13 その他.....	29
4.13.1 差分表示.....	29
4.13.2 ダイアグラム印刷.....	29
4.13.3 自動整列.....	29
4.13.4 検索・置換.....	29
5 パターン.....	30
5.1 概要.....	30
5.2 表記の変更.....	30
5.3 パターンの構造.....	30
5.4 パターンの追加.....	30
5.4.1 ダイアグラムへの追加.....	30
5.4.2 ノードへの追加.....	31
5.4.3 パターンの追加処理.....	31
6 パラメータ.....	32
6.1 概要.....	32

6.2 パラメータ定義および値の設定.....	32
6.3 パラメータ値の参照.....	33
6.4 パラメータの一覧表示.....	34
6.5 パラメータ定義および値の格納.....	34

1 はじめに

1.1 概要

本書は、独立行政法人 科学技術推進機構(以下、JST)が行う戦略的創造研究推進事業の研究領域である「実用化を目指した組込みシステム用ディペンダブル・オペレーティングシステム」(以下、DEOS プロジェクト)において開発中の、Dependability cases(以下、D-Case)の作成を支援するツール「D-Case Editor」の拡張機能に関する基本設計書である。

1.2 用語の定義

名称	意味
ダイアグラム	(D-Case を)図表化したもの。
D-Case 文書	D-Case およびモジュールを表すファイル。下記の GMF ダイアグラム情報ファイルおよび GMF モデル情報ファイルの 2 ファイルからなる。
GMF ダイアグラム情報ファイル	D-Case もしくはモジュールの各要素(ノードやリンク)の位置や大きさ、色などの情報を表すファイル。 「D-Case 名もしくはモジュール名.dcase_diagram」のファイル名で記録される。
GMF モデル情報ファイル	D-Case もしくはモジュールの各要素の論理的な構造を表すファイル。 「D-Case 名もしくはモジュール名.dcase_model」のファイル名で記録される。
属性	ノードやリンクがそれぞれ固有に持つ性質。 Name, Desc, Attachment, Userdef001～016 などがある。
パレットビューア	Eclipse 上で動作するグラフィカルエディタ(D-Case Editor)内で、ノードやリンクを選択するツールを提供する部分。
ビュー	Eclipse 内で何らかの情報を提供する部分。通常はタブ形式でいずれか 1 つの情報が表示されている。
プリファレンスストア	Eclipse でプラグインの設定を保存するための領域。ワークスペース(作業領域)内に設けられ、プラグイン毎に分けて記録される。

1.3 関連文書

- 発注仕様書
- D-Case Editor 機能仕様書
- D-CASE エディタ データ仕様書
- D-Case 入門
- GSN COMMUNITY STANDARD VERSION1
http://www.goalstructuringnotation.info/documents/GSN_Standard.pdf
- Eclipse3.4 プラグイン開発 徹底攻略 毎日コミュニケーションズ (ISBN978-4-8399-2972-5)
- Eclipse Documentation
<http://www.eclipse.org/documentation/>

- Graphical Modeling Framework – Eclipsepedia
http://wiki.eclipse.org/Graphical_Modeling_Framework
- Eclipse Modeling Framework – Eclipsepedia
http://wiki.eclipse.org/Eclipse_Modeling_Framework
- Graphical Editing Framework – Eclipsepedia
<http://wiki.eclipse.org/GEF>
- Java Platform Standard Edition 7 Documentation
<http://docs.oracle.com/javase/7/docs/index.html>

2 拡張機能概要

D-Case Editor は、DEOS プロジェクトにおいて開発中の、D-Case の作成を支援するためのツールである。オープンソースの統合開発環境である「Eclipse」のプラグインであり、基本的な機能はすでに実装されている。しかし、企業などで実際に開発されているシステムの構造は複雑であり、記述される D-Case は大きくなる傾向がある。そのため、実際に本ツールを利用する際には、以下の機能が必要となる。

- 大規模な D-Case を単一のダイアグラムで管理することが困難なとき、モジュールとして分割することで、ダイアグラムを統合・簡素化し、管理しやすくする
- 優れた部分 D-Case をパターンとして登録し、パターンを利用することで、D-Case の記述を行いやすくする
- パラメータのスコープをより柔軟にすることで、D-Case の汎用化を促進し、パターンの再利用性を高める

上記のそれぞれの拡張機能の外部仕様を、以降で述べる。

3 ノードとリンク

D-Case の記法は GSN を拡張したものである。GSN で規定されたノードおよびリンクは、D-Case Editor で実装されている。しかし、D-Case で規定されたノードおよびリンクは、一部実装されていない。そのため、以下の変更を行う。

3.1 ノード

現在の D-Case Editor で実装されている「システム(System)」および「ポリシー(Policy)」ノードを削除する。そして、下記のノードを追加する。

1. アクション(Action)

「証拠(Evidence)」ノードのサブクラスである。運用時のシステムから得られるログ情報などによるエビデンスである。

2. 外部接続(External)

「モジュール(Module)」ノードのサブクラスである。あるモジュールを参照するとき、そのモジュールが現時点の範囲外(外部システム)である場合に用いる。

3. パラメータ(Parameter)

「前提(Context)」ノードのサブクラスである。パラメータの定義、型、値を記述する。

いずれも、ノードの形状、表示する内容、属性入力ダイアログで編集可能な属性などは、スーパークラスと同じである。アイコンおよびノード名のプレフィックスは、それぞれに適したものを使用する。図 1 に例を示す。



図 1:アクション、外部接続およびパラメータの形状

ただし、パラメータノードでは、記述されているパラメータの情報を、以下の形式でノード内に表示する。

パラメータ名 1: 型名=値

パラメータ名 2: 型名=値

...

3.2 リンク

d*表現で使用する、責任委譲関係を表すリンクを追加する。(図 2)。

責任委譲関係を表すリンクの詳細については 4.9.2、d*表現の詳細については、4.9.3 で述べる。

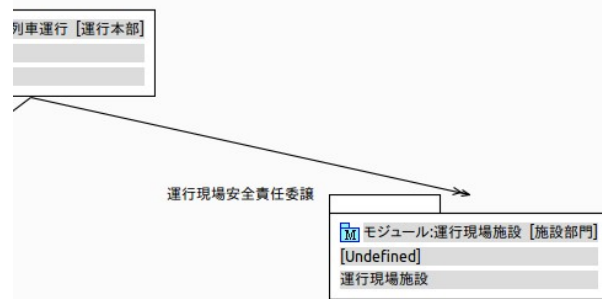


図 2:責任委譲関係を表すリンク

4 モジュール

4.1 概要

大規模な D-Case を効率的に管理するため、モジュール機能を追加する。あるノードをルートとするサブツリーを「モジュール」というひとつのまとまりにして、単一の「Module ノード」に置き換えられるようにする。Module ノードは、通常では単一のノードで表示されるが、展開することによりモジュール内部のノードを表示し、確認することもできる。モジュールは D-Case 文書として保存し、参照元の D-Case とは別のダイアグラムで編集する。

また、モジュールのルートノード以外を参照する「Away Goal」の機能を追加する。

Module ノードおよび Away Goal ノードからの参照を許可するかどうかは、モジュールおよびノードの public/private 属性で判断する。public 属性を持つモジュールおよびノードのみ参照可能とする。

4.2 モジュールの構造

4.2.1 ファイルの格納場所

モジュールは、従来の D-Case と同様、GMF ダイアグラム情報ファイル(dcase_diagram)および GMF モデル情報ファイル(dcase_model)の形式で保存される。モジュールのファイル名は、「モジュール名.dcase_diagram」および「モジュール名.dcase_model」とし、参照元のノードを含む D-Case 文書が格納されるフォルダに格納する(図 3)。

```
main.dcase_diagram
main.dcase_model
Module1.dcase_diagram
Module1.dcase_model
Module2.dcase_diagram
Module2.dcase_model
⋮
```

図 3: D-Case 文書の保存場所

このため、D-Case は、あるフォルダ直下にある D-Case 文書群からなる。

また、トップのノードを含む D-Case 名を「main」とし、ファイル名を「main.dcase_diagram」および「main.dcase_model」とする。ただし、main もモジュールの一種として扱う。

4.2.2 モジュールの参照

Module ノードおよび Away Goal ノードが参照するモジュールやモジュール内のノードは、属性「Attachment」に記録する。また、参照されるモジュールやノードは、参照元のノードを属性「Userdef011」に記録する。図 4 に例を示す。

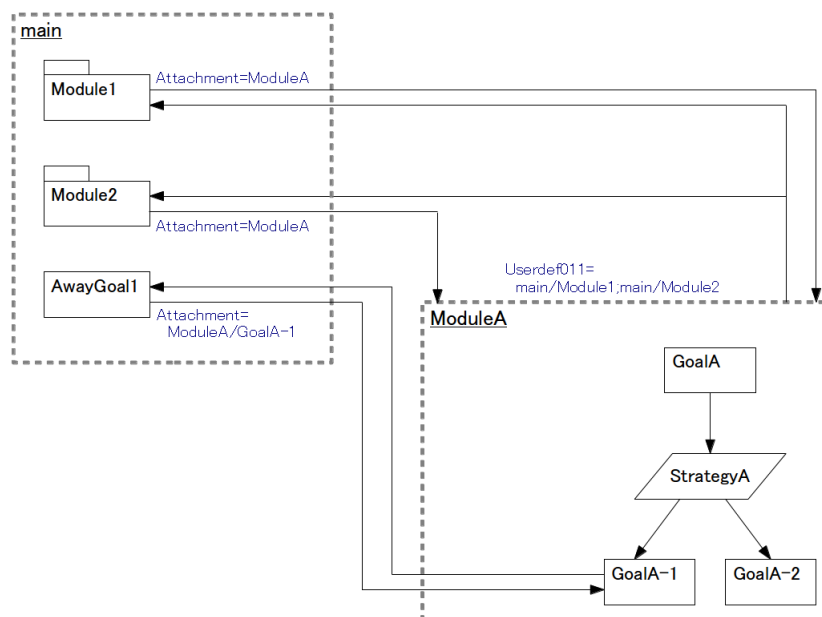


図 4: ノードとモジュールの参照の例

4.3 モジュールの作成

4.3.1 Module ノードの作成

Module ノードを作成するには、他のノードと同様、パレットビューアの「GSN Nodes」にある「Module」を選択し、ダイアグラム上で設置場所を指定する。(モジュールを指定する手順は 4.4.1 に記載)

4.3.2 既存ノードのモジュール化

ダイアグラムにある複数のノードをモジュール化する手順は、以下の通りである。

モジュール化するサブツリーのルートノードを選択して右クリックしたとき、コンテキストメニューに「Create Module」を表示する(図 5)。これを選択したとき、以下の処理を行う。

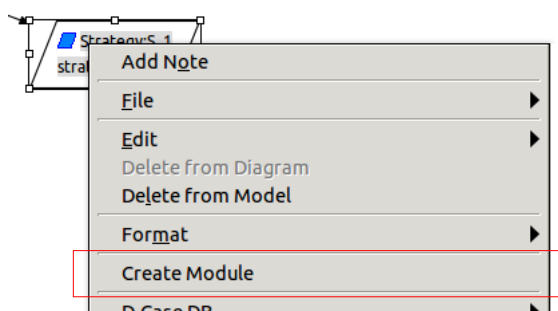


図 5: モジュール化のためのメニュー

1. 選択されたノードをルートとするサブツリーをモジュール化の対象とするため、選択状態にする。
ルートノードから再帰的に子孫ノードに向かうリンクをたどり、ルートノードを含めた子孫ノードを選択状態にする。
2. 1.で選択状態にしたサブツリーがグループになる(子孫ノードをたどる過程でルートノードに到達する場合、エ

ラードログに以下を出力して、モジュール化の処理を終了する。

Not a single tree.

3. モジュール名を入力するためのダイアログを表示し(図 6)、モジュール名を得る。キャンセルボタンが押された場合は、モジュール化の処理を終了する。

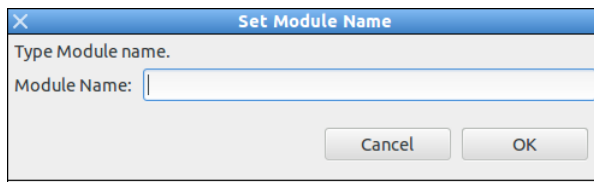


図 6: モジュール名設定ダイアログ

すでに同名の D-Case 文書が存在する場合は、再度ダイアログを表示する。

4. 1.で選択された単一のサブツリーの内容の D-Case 文書を、現在編集中のダイアグラムと同フォルダ(以下、カレントフォルダ)に、3.で入力されたモジュール名を用いた以下のファイル名で保存する。

モジュール名.dcase_diagram

モジュール名.dcase_model

ファイルシステムの問題などにより上記ファイルを作成できない場合は、下記のエラーログを出力して、モジュール化の処理を終了する。

モジュール名: create failed.

5. 1.で選択したサブツリーを、現在のダイアログから削除する。
ルートノード以外で、サブツリー以外から参照されているノードがある場合は、そのノードを Away Goal ノードとして残す。Away Goal ノードに関しては、4.4.2 を参照されたい。
6. Module ノードを追加する。Name 属性および Attachment 属性に、3.で入力したモジュール名を設定する。また、1.のサブツリーがあるノードの子ノードだった場合は、Module ノードをそのノードの子ノードとする。
7. 参照元の Module ノードを設定する。4.で生成したモジュールの Argument の属性「Userdef011」に、参照元の Module ノードのパスを設定する。
モジュール名/Module ノード名
8. 責任属性をコピーする。4.で生成したモジュールの Argument の属性「Userdef012」に、自身のモジュールの Argument の Userdef012 属性の値を設定する。(責任属性については 4.9 に記載)
9. 親モジュールを設定する。4.で生成したモジュールの Argument の属性「Userdef013」に、参照元の Module ノードが所属するモジュール名を設定する。

4.4 モジュールの参照

4.4.1 Module ノードの参照先の指定

既存の Module ノードの参照先モジュールを設定する手順は、以下の通りである。

Module ノードを選択して右クリックしたときに表示されるコンテキストメニューの「Attachment」に、「Select from Module...」を表示する(図 7)。



図 7: モジュールを選択するためのメニュー

これを選択すると、カレントフォルダにある D-Case 文書から、public 属性を持つモジュールか、自身が親モジュールである(Userdef013 属性に自モジュールが設定された)モジュールの一覧を作成し、メニューとして表示する。これらのいずれかを選択すると、Attachment 属性に、選択したモジュール名を設定する。

また、選択したモジュールの Argument の属性「Userdef011」に、参照元の Module ノードとなる下記を設定する。すでに設定されている(別のノードから参照されている)場合は、「;」を追加した上で下記を追加設定する。

モジュール名/Module ノード名 (新規の場合)

既存の設定;モジュール名/Module ノード名 (設定されている場合)

そして、Module ノード内にモジュール名を表示する(図 8)。

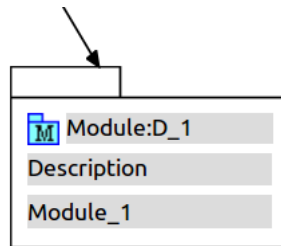


図 8: Module ノードの表示

4.4.2 Away Goal

Away Goal は、モジュール内のルートでないノードを参照するためのノードであり、GSN で規定されている。

D-Case Editor では、モジュール内のノードを参照する Goal ノードを、Away Goal ノードとして扱う。

既存の Goal ノードの参照先ノードを設定し、Away Goal ノードとする手順は、以下の通りである。

4.4.1 の場合と同様、Goal ノードを選択して右クリックしたときに表示されるコンテキストメニューの「Attachment」に、「Select from Module...」を表示する(図 7)。これを選択すると、カレントフォルダにある D-Case 文書から、public 属性を持つノード名の一覧を作成し、「モジュール名/ノード名」の形式でメニューとして表示する。これらのいずれかを選択すると、Attachment 属性に、選択したノードのパスを設定する。パスの形式は以下の通りである。

モジュール名/ノード名

また、選択したノードの Argument の属性「Userdef011」に、参照元の Away Goal ノードとなる下記を設定する。
すでに設定されている場合は、「;」を追加した上で下記を追加設定する。

モジュール名/Away Goal ノード名 (新規の場合)

既存の設定;モジュール名/Away Goal ノード名 (設定されている場合)

Away Goal ノードの表示を、通常の Goal ノードと区別するため、図 9 のように緑の背景色とする。

また、Module ノードと同様、Attachment 属性の内容をノード内に表示する。

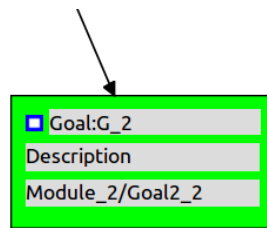


図 9: Away Goal ノード

同様に、Away Goal ノードから参照されているノードの表示を、図 10 のように背景を橙色とする。



図 10: Away Goal から参照されたノード

4.5 モジュール・サブツリーのコピー

既存のモジュールやサブツリーを利用する際、参照ではなくコピーする機能を追加する。

4.5.1 モジュールのコピー

モジュールのコピーを作成し、コピーしたモジュールを Module ノードから参照する手順は、以下の通りである。

Module ノードを選択して右クリックしたときに表示されるコンテキストメニューの「Attachment」に、「Copy Module...」を表示する(図 11)。

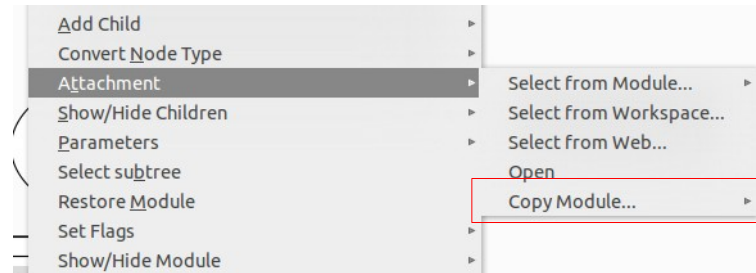


図 11: モジュールをコピーするためのメニュー

これを選択すると、4.4.1 と同様、モジュールの一覧を表示する。これらのいずれかを選択すると、4.3.2 の図 6 のモジュール名設定ダイアログを表示し、入力されたモジュール名でコピーを作成する。そして、Module ノードからコピーしたモジュールを参照する。個々の手順の詳細は、4.3.2 の通りである。

4.5.2 サブツリーのコピー

public 属性のノードをルートとするサブツリーをコピーし、Goal ノードの下に追加する手順は、以下の通りである。

Module ノードを選択して右クリックしたときに表示されるコンテキストメニューの「Attachment」に、「Copy Subtree...」を表示する(図 12)。

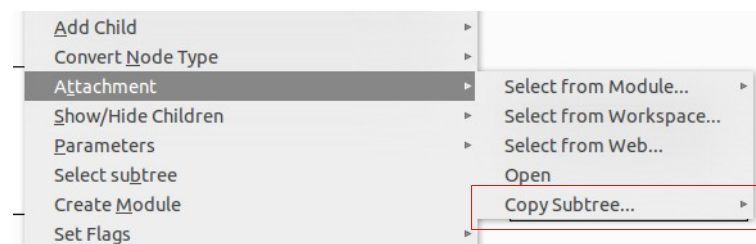


図 12: サブツリーをコピーするためのメニュー

これを選択すると、4.4.2 と同様、public 属性を持つノードの一覧を表示する。これらのいずれかを選択すると、そのノードをルートとするサブツリーをコピーし、Goal ノードの下に追加する。その際、コピーされるノードの Desc Format String 属性内で使われるパラメータのうち、サブツリー内で値が設定されていないものについては、コピー元で設定された値に変換する。

4.6 public/private 属性

public/private 属性を設定する手順は、以下の通りである。

Argument もしくはノードを右クリックしたとき、コンテキストメニューに「Set Flags」を表示する(図 13)。これを選択すると、「Private」および「Public」を表示する。これらを選択すると、ノードに public/private 属性を設定する。



図 13: public/private 属性設定のためのメニュー

public/private 属性は、ノード属性「Userdef010」を使用して実現する。Userdef010 属性に「P」の文字が含まれている場合は public、含まれていない場合は private とする。デフォルトでは private が設定されているとする。

このため、「Properties」ビューの Userdef010 の属性の表示を、「Flags」とする。

4.7 モジュールの解除

Module ノードを削除し、モジュールのツリーを直接展開する手順は、以下の通りである。

Module ノードを選択して右クリックしたとき、コンテキストメニューに「Restore Module」を表示する(図 14)。これを選択したとき、以下の処理を行う。

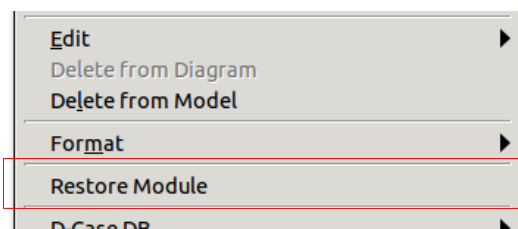


図 14: モジュール解除のためのメニュー

1. Module ノードを削除する。
2. モジュール化されていたツリーを、Module ノードが存在した箇所に(子ノードだった場合は親ノードの下に、ダイアグラム直下だった場合はダイアグラムに)追加する。
3. モジュールの Argument の Userdef011 属性から、削除した Module ノードのパスを削除する。
4. 2で追加したツリーを選択状態とする。

なお、ここでは、参照先モジュールの D-Case 文書の削除は行わない。(削除の手順は 4.10 に記載)

4.8 モジュールの展開表示

Module ノードおよび Away Goal ノードが参照するモジュールの内容(サムネイル)を、そのノードの中に表示する(あるいは表示をやめる)手順は、以下の通りである。

Module ノードを右クリックしたとき、コンテキストメニューに「Show/Hide Module」を表示する(図 15)。これを選択すると、「Show Module」および「Hide Module」を表示する。

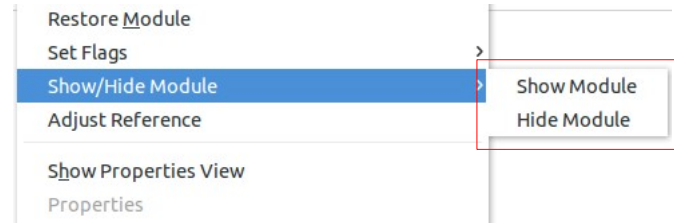


図 15: モジュールを展開するためのメニュー

「Show Module」を選択すると、Module ノードの中にモジュールの内容を表示する(図 16)。すでにモジュールの内容を表示している場合は、何も実施しない。

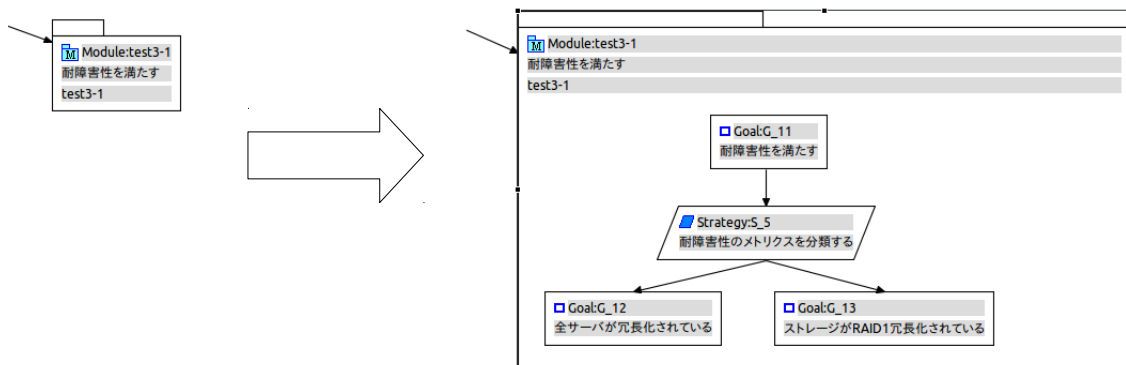


図 16: モジュールの展開

「Hide Module」を選択すると、モジュールの内容を表示している場合は、もとの Module ノードの表示に戻す。

Away Goal ノードの場合も、同様にコンテキストメニューを表示し、同様に表示・非表示の処理を行う。ただし、「Show Module」を選択したときには、参照先のノードをルートノードとするサブツリーを表示することになる。

4.9 責任属性

責任属性の設定および表示に関する手順を、以下に示す。

4.9.1 責任属性の追加

各ノードに責任属性を設定できるようにする。ノードをダブルクリックしたときに表示される属性入力ダイアログに、責任属性を入力する項目を追加する(図 17)。

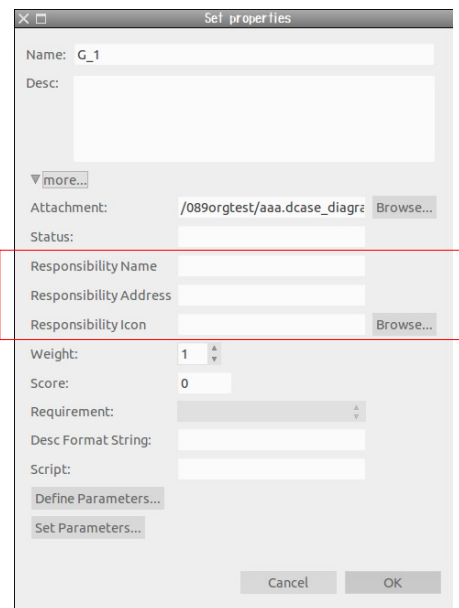


図 17:属性入力ダイアログ

項目はそれぞれ、責任属性の名前、メールアドレスおよびアイコン画像のパスである。アイコン画像は、Browse ボタンを押すことで、ファイルダイアログを介して

Ok ボタンが押されたとき、入力された文字列を属性「Userdef012」に、下記の形式で設定する。

責任属性の名前; メールアドレス; アイコン画像のパス

そして、ノード名の右側に、責任属性のアイコン画像を表示する(図 18)。

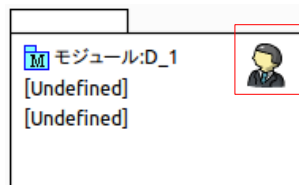


図 18:責任属性アイコンの表示

Module ノードの場合は、参照先のモジュールの責任属性と同期させる。参照先モジュールを設定する際に、モジュールの責任属性がすでに設定されていれば、Module ノードの Userdef012 属性に設定する。

モジュールの責任属性は、Argument の属性「Userdef012」に設定する。Argument を選択して右クリックしたときに表示されるコンテキストメニューに、「Set Responsibility」を表示する(図 19)。

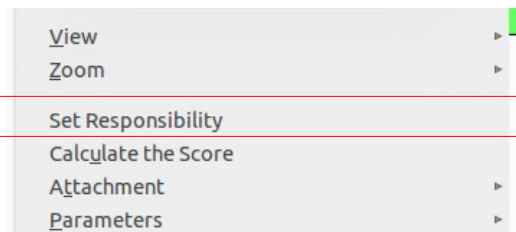


図 19:責任属性を設定するためのメニュー

これを選択すると、責任属性を入力するためのダイアログを表示する(図 20)。



図 20:責任属性入力ダイアログ

「OK」ボタンが押されたら、入力された値を Argument の Userdef012 属性に、前述と同形式で設定する。

4.9.2 責任委譲関係のリンク

4.9.3 で述べる d*表現で使用される、モジュール間の責任委譲関係を表すためのリンクである。形状は、3.2 で記述した通り。

モジュール間の関係から自動的に生成される場合と、ユーザによって追加される場合がある。後者によって生成されたリンクは、線を青色にすることで、前者と区別できるようにする(図 21)。

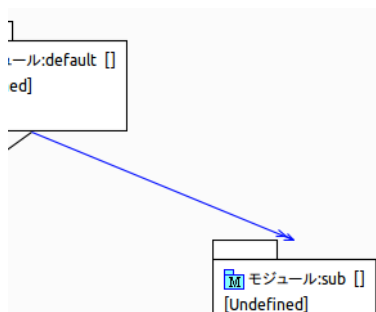


図 21:手動で作成したリンク

責任委譲関係を表すための D-Case を、Attachment 属性に設定できるようにする。リンクを右クリックしたときに表示されるコンテキストメニューに「Attachment」を表示する(図 22)。

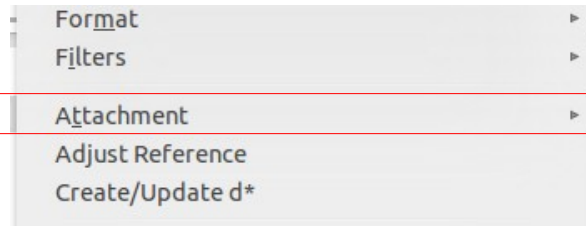


図 22:責任委譲の D-Case を設定するメニュー

これを選択すると、カレントフォルダにある D-Case 文書から、モジュール名一覧を作成し、メニューとして表示する。これらのいずれかを選択すると、Attachment 属性に選択したモジュール名を設定する。

さらに、両端のモジュールに責任属性が設定されており、責任属性が同じ場合は「契約(Contract)」を表すアイコンを、責任属性が異なる場合は「責任契約(Responsibility Contract)」を表すアイコンを、モジュール名の左横に表示する。

リンクに Attachment を設定した場合、図 2 のように、Desc 属性ではなく、Attachment 属性を表示する。また、図 22 のメニュー「Attachment」で表示するモジュール名一覧の末尾に、「New...」を追加する(図 23)。



図 23:モジュールを新規作成するためのメニュー

これを選択したとき、4.3.2 の図 6 のモジュール名設定ダイアログを表示する。そして、入力された名前のモジュールを新規に作成し、リンクの Attachment 属性にその名前を設定する。4.3.2 と同様、キャンセルボタンが押されたときや、同名の D-Case 文書がすでに存在するときは、同じ処理を行う。

4.9.3 d*表現

モジュール間の関係を表す「d*表現」をダイアグラムとして表示する機能を追加する。

モジュールの Argument を右クリックしたとき、コンテキストメニューに「Create/Update d*」を表示する(図 24)。

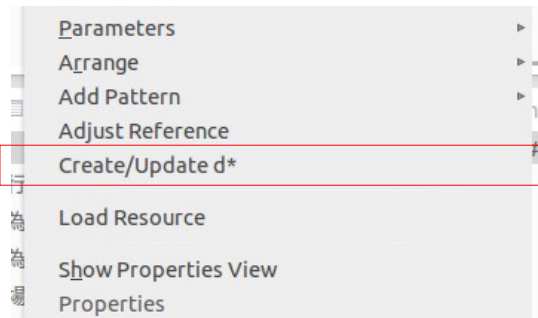


図 24:d*表現を表示するためのメニュー

これを選択したとき、図 25 のような d*表現のダイアグラムを追加する。
 すでに存在する場合は、ダイアグラムの内容を最新の状態に更新する。

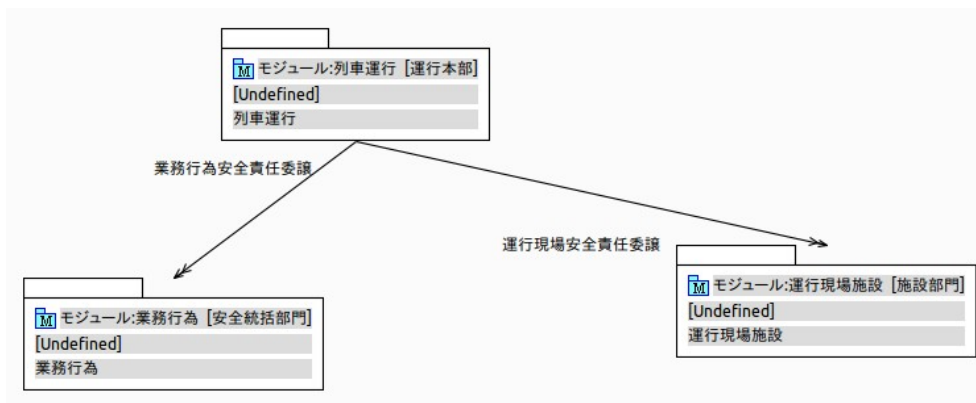


図 25:d*表現の例

d*表現のダイアグラムを表示する手順は、以下の通り。

1. ダイアグラムを新規に作成する。D-Case 文書がすでにある場合は、そのダイアグラムをオープンする。
 D-Case 文書のファイル名は以下の通り。
 main.dstar_diagram
 main.dstar_model
2. D-Case 文書がすでにある場合は、責任委譲関係のリンクに設定された Attachment 属性の値を記録しておく。
3. カレントフォルダにあるモジュールを、Module ノードとして追加する。各属性を以下のように設定する。
 パラメータなどの処理は行わないため、下記以外の属性は設定しない。
 Name および Attachment : モジュール名
 Desc : Argument の責任属性の名前およびメールアドレス
 Userdef012 : Argument の Userdef012 属性の値
4. モジュール内に、他のモジュールを参照する Module ノードもしくは Away Goal ノードがある場合、自 Module ノードと参照先の Module ノードを、4.9.2 で示した責任委譲関係のリンクで接続する。

5. 2.で保存しておいた Attachment 属性を、対応するリンクの Attachment 属性に設定する。

当ダイアグラムは、D-Case と同等のダイアグラムのため、すべてのノードやリンクの追加、削除、編集等を自由に行うことができる。しかし、当ダイアグラムでは、責任委譲関係を表すリンクの追加と、そのリンクに Attachment を設定することだけを想定する。そのため、d*表現の D-Case 文書がすでに存在する場合は、手動で追加された責任委譲関係のリンクと、リンクに設定された Attachment 属性の値のみ保持し、他の情報は破棄して、新たに作成し直す。

4.10 モジュールの管理

「Modules」ビューを表示してモジュールの管理を行う手順は、以下の通りである。

「Window」メニューの「Show View」で「Other...」を選択し、「D-Case Editor」を展開すると、「Modules」が表示される(図 26)。

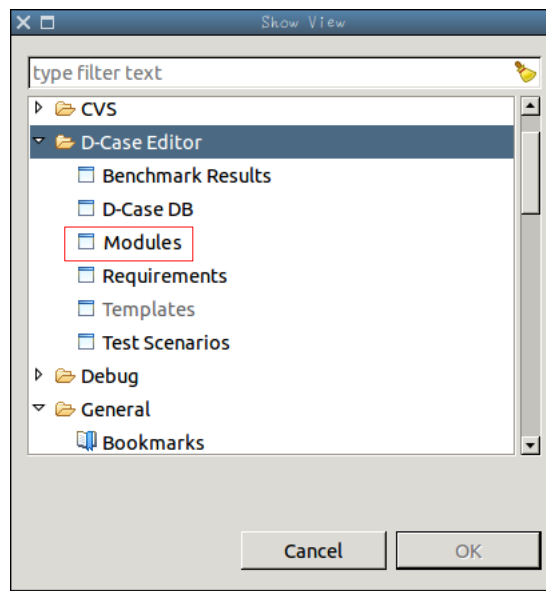


図 26: モジュールビュー表示の選択

これを選択すると、Modules ビューを表示する(図 27)。

Modules ビューでは、現在表示されているダイアグラムが属する D-Case のモジュールおよび public ノードと、モジュール内のノード数、それぞれのリンク数(Userdef011 属性の「;」の個数+1)、参照元のノード(Userdef011 属性値)を表形式で表示する。

Name	Node#	Link#	Reference
main	9	0	
main/G_1	9	1	test5/G_9
main2	6	1	main/D_1
main2/G_5	6	0	

図 27: Modules ビュー

ここで、モジュールを選択して、右上にあるツールバーの FORWARD アイコン(矢印)を押下すると、指定したモジュールをオープンする。また、モジュールを選択して、ツールバーの DELETE アイコン(×印)を押下すると、モジュールおよび public ノードのリンク数がいずれも 0 の場合に、指定したモジュールの D-Case 文書を削除する。

また、属性を直接編集するなどして、モジュール間の参照が双方向にならなくなることがある。

これを修正するには、Attachment を右クリックしたとき、コンテキストメニューに「Refresh」を表示する(図 28)。これを選択すると、以下の処理を行う。

1. モジュールもしくは public ノードが Argument に設定されていて、モジュールもしくは public ノードがすでに存在しない場合、Attachment の設定を削除する。
2. モジュールもしくは public ノードが Argument に設定されていて、参照されている Argument もしくは public ノードの Userdef011 にそのノードが設定されていない場合、Attachment の設定を削除する。
3. Userdef011 に登録されている各ノードに対して、そのノードがすでに存在しない場合、そのノードの設定を削除する。
4. Userdef011 に登録されている各ノードに対して、そのノードの Attachment に自ノードが登録されていない場合、そのノードの設定を削除する。
5. 参照が双方向であるとき、参照先の責任属性(Userdef012 属性)と、参照元の責任属性が異なる場合、参照先の Userdef012 属性の値を、参照元の Userdef012 属性に設定する。

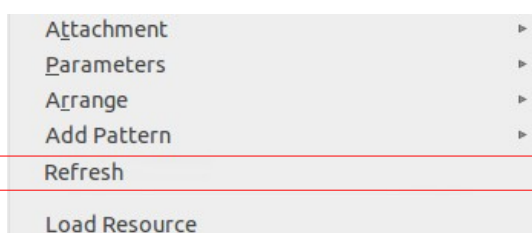


図 28: モジュールの参照を同期させるためのメニュー

4.11 ノード一覧の出力

モジュールにどのようなノードが含まれているか確認できるよう、ノードの一覧をテキストファイルに出力する手順は、以下の通りである。

「D-Case」メニューの「Convert D-Case File」を選択したとき、「Convert GMF To Text」を表示する(図 29)。

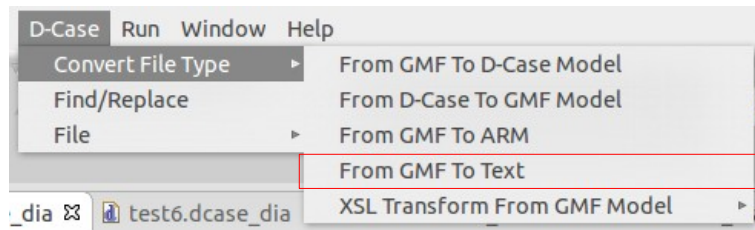


図 29: ノード一覧を出力するためのメニュー

これを選択すると、テキストファイルに出力するためのウィザードを表示する(図 30)。

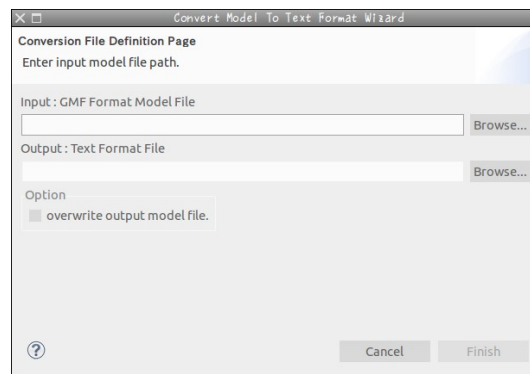


図 30: ノード一覧をテキストファイルに出力するためのウィザード画面

ここで、入力となる GMF モデル情報ファイルと、出力となるテキストファイルを入力し、「Finish」ボタンを押下すると、ノードの種類ごとに分けた、下記のフォーマットのテキストファイルを出力する。

[ノード名]

"[Name](#)", "[Desc](#)", "[Attachment](#)"

...

[ノード名]

"[Name](#)", "[Desc](#)", "[Attachment](#)"

...

4.12 日本語化

Eclipse プラグインの国際化機能を使用して、ノード名の日本語化を行う。

ノード名は、下記のように日本語に変換する。

ノード名(英語)	ノード名(日本語)
Goal	ゴール
Context	前提
Strategy	戦略
Evidence	証拠
Justification	正当化
Assumption	仮定
Undeveloped	未達成
Module	モジュール
Contract	契約
Monitor	モニタ
Action	アクション
External	外部接続
Parameter	パラメータ

また、各所で表示されるメッセージは、下記のように日本語に変換する。

場所	メッセージ(英語)	メッセージ(日本語)
「D-Case」メニュー	Convert File Type	フォーマットの変換
	From XXX to YYY model	XXX から YYY への変換
	Find/Replace	検索/置換
	Copy To	次へコピー
	Rename	名前変更
	Compare To	比較
	Switch Source and Target	リンク元とリンク先の入れ替え
	Arrange Direction	整列
	Vertical	垂直方向
	Horizontal	水平方向
パレットビューア	GSN Nodes	GSN ノード
	Create GSN Nodes	GSN ノードの生成
	D-Case Nodes	D-Case ノード

	Create D-Case Nodes	D-Case ノードの生成
	Links	リンク
	Create Links	リンクの生成
フォーマット変換 Wizard	GMF Format Model File	GMF モデル情報ファイル
	D-Case Format Model File	D-Case モデル情報ファイル
	overwrite output model file	出力ファイルの上書きを許可
	Enter input model file path	入力ファイルを指定してください
	Enter output model file path	出力ファイルを指定してください
	The output file is the same as an input file	出力ファイル名が入力ファイル名と同じです
ARM 変換 Wizard	ARM Format File	ARM 形式ファイル
XSL 変換 Wizard	XSL Transform - {0}	XSL 変換 - {0}
テキスト変換 Wizard	Text Format File	テキストファイル
Diagram 作成 Wizard	Create new diagram based on {0} model content	{0}モデルの新しいダイアグラムの作成
	Select diagram root element	ダイアグラム・ルート要素の選択
	Diagram root element	ダイアグラム・ルート要素
	Select semantic model element to be depicted on diagram	ダイアグラムのセマンティックモデルの選択
	Diagram root element is not selected	ダイアグラム・ルートが選択されていません
	Invalid diagram root element is selected	不正なダイアグラム・ルートが選択されています
	Initializing diagram contents	ダイアグラムの初期化
	Incorrect model object stored as a root resource object	不正なモデルオブジェクトがルート・リソースオブジェクトとして保存されています
D-Case 作成 Wizard	New D-Case Diagram	新規 D-Case ダイアグラム作成
	Create D-Case Diagram	D-Case ダイアグラムファイルの作成
	Select file that will contain diagram model	D-Case ダイアグラムファイルを入力してください
	Create D-Case Domain Model	D-Case モデル情報ファイルの作成
	Select file that will contain domain model	D-Case モデル情報ファイルを入力してください
	File name should have {0} extension	{0} は適切ではありません
Parameter 作成 Wizard	Create a New Parameter	新しいパラメータの設定
	Create a new parameter item	新しいパラメータの設定
	Length	長さ
	Digit	数字
	Incremental	インクリメンタル
	Input the double property	double 型の設定
	Range	範囲

	The specified item is already exists. The item must be unique	その名前のパラメータは既に存在しています	
	The enumeration must have at least one item	enum 型は少なくとも一つの値を持たなくては いけません	
	Set up the enumeration items	enum 型の設定	
	Input the integer property	int 型の設定	
	The specified Name is already used. Name must be unique	その名前は既に存在しています	
	Input the raw property	raw 型の設定	
	Input the string property	string 型の設定	
	"{0}" is a reserved word	"{0}"は予約語で、使えません	
	Detected an illegal character in {0}	不正な文字が使われています{0}	
Parameter 編集 Wizard	{0} is not correct	{0} は適切ではありません	
	{0} is empty	{0} は空です	
	{0} is invalid	{0} は不正です	
	Edit the Parameter	パラメータの編集	
	Edit the parameter item	パラメータの編集	
属性入力 Dialog	Set properties	プロパティの設定	
	Attachment	添付	
	Status	状況	
	Browse	参照	
	Score	スコア	
	Weight	重み	
	Requirement	要件	
	Desc Format String	パラメータを使った文章	
	Script	スクリプト	
	Node Link	ノードリンク	
	Is Normal	ノーマル	
	Stakeholder	ステークホルダー	
	Risk Analysis	リスク分析	
	,URL is empty	URL が空です	
	{0} is invalid URL	{0} は不正な URL です	
	Bookmark	ブックマーク	
	{0} not supported protocol	{0} は未サポートのプロトコルです	
	検索 Dialog	Find	検索
		Replace with	置換
The string was not found		見つかりませんでした	
1 match replaced		1件置換しました	

	%d matches replaced	%d 件置換しました
Copy To Dialog	Copy D-Case file	D-Case ファイルのコピー
	Copy D-Case files to an other location	別の場所にコピー
	Select D-Case Diagram File	D-Case ダイアグラムファイルの選択
Node Link 選択 Dialog	Select Node Link	ノードリンクの選択
Bookmark 設定 Dialog	Name	名前
	New name	新しい名前
Confirm	Some properties will be moved to Userdef008. Are you sure to continue?	いくつかの情報が Userdef008 へ異動しますがよろしいですか？
Patterns ビュー	Add Pattern	パターンの追加
	Add pattern to editing D-Case diagram	パターンをダイアグラムに追加する
Requirements ビュー	Requirement	要件
	Scroll to Goal	ゴールへスクロール
	Scroll to Requirement	要件へスクロール
	Refresh	リフレッシュ
Compute nodes ビュー	Compute nodes	ノード数のカウント
	All nodes	ノード数合計
	copy	コピー
コンテキストメニュー	Validate	検証
	Save As Eps File	EPS ファイルとして保存
	Selected file is already exist. will you overwrite this file?	選択した名前のファイルはすでにあります。上書きしますか？
	Initialize dcase_diagram diagram file	D-Case ダイアグラムの初期化
	Add Pattern	パターンの追加
	Set Parameters	パラメータの設定
	Create/Update d*	d*表現の作成/更新
	Select from Module	モジュールから選択
	Select from Workspace	ワークスペースから選択
	Select from Web	.Web から選択
	Calculate the Score	スコアの計算
	Set Responsibility	責任属性の設定
	Convert Node Type	ノードの種類を変更
	Attachment	添付
	Hide Children	子の非表示
	Show/Hide Children	子の表示/非表示
	Create a New Node	ノードの追加
	Select subtree	サブツリーの選択
	Create Module	モジュールの作成

	Restore Module	モジュールの展開
	Set Flags	フラグの設定
	Private	プライベート
	Public	パブリック
	Show/Hide Module	モジュールの表示/非表示
	Add Pattern to node	パターンの追加
	Set Requirements	要件の設定
	Risk Analysis	リスク分析
	Open with Web Browser	Web ブラウザーで開く
	Set Sibling Order	サブゴールの兄弟順序の設定
	Node Link	ノードリンク
	Open with D-Case Editor	D-Case エディタで開く
その他	Select domain model element	ドメインモデル要素の選択
	Available domain model elements	利用可能なドメインモデル要素
	incoming links	入力方向のリンク
	outgoing links	出力方向のリンク

4.13 その他

下記については、Module ノードを単一のノードとして扱うことで、基本的には現在の D-Case Editor と同等に処理を行うこととする。

4.13.1 差分表示

Module ノードは単一のノードとして扱い、モジュールの内部は比較対象としない。

4.13.2 ダイアグラム印刷

現在表示されている状態で印刷される。(モジュールを展開している場合は、展開した状態で印刷される。)

4.13.3 自動整列

Module ノードは、他のノードと同様に扱われる。

4.13.4 検索・置換

Module ノードは単一のノードとして扱うため、モジュールの内部は検索対象としない。

5 パターン

5.1 概要

現在の D-Case Editor には、「テンプレート」を追加する機能が実装されている。ただし、現在のテンプレート機能では、ダイアグラムのどの位置に追加されるかがわからない。これを、ダイアグラム上の所望する位置に追加できるようにする。さらに、指定したノードの子ノードとして追加できるようにもする。

5.2 表記の変更

他の文献では、当機能を「パターン」と呼ぶことが多いため、今後は「パターン」という表記で統一する。そのため、下記で「Template」と記載されている箇所を、「Pattern」に変更する。

ただし、パターンが格納されているプロジェクトの名称「D-CaseTemplate」は、混乱を避けるため、変更しない。

- テンプレートビューの名称
(Windows メニューの「Show View」で表示されるビューの名称(Templates)を含む)
- Templates ビューのプルダウンメニュー内の「Add Template」
- ノードを右クリックしたコンテキストメニュー「Add Child」内の「Add a Template」

5.3 パターンの構造

パターンは、単一のツリーであり、かつ単一の D-Case 文書であることを前提としている。そのため、内部に Module ノードおよび Away Goal ノードを含まない。

5.4 パターンの追加

5.4.1 ダイアグラムへの追加

パターンビュー(旧テンプレートビュー)からパターンを追加する際の処理は、現在と同様である。ただし、パラメータの処理があるため、5.4.3 の処理は実施する。

ダイアグラム上の指定した位置に追加する手順は、以下の通りである。

ダイアグラムを右クリックしたとき、コンテキストメニューに「Add Pattern」を表示する(図 31)。ここで、登録されているパターンのうち、単一のツリーのもをコンテキストメニューに一覧表示する。パターンのいずれかを選択すると、右クリックした位置にパターンを追加する(詳細は 5.4.3 に記載)。

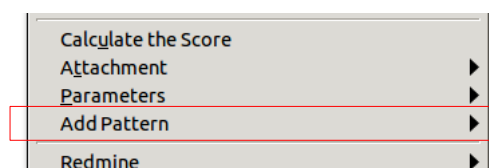


図 31: ダイアグラムにパターンを追加するためのメニュー

5.4.2 ノードへの追加

指定したノードの下にパターンを追加する手順は、以下の通りである。

ノードを選択して右クリックしたときに表示されるコンテキストメニューの「Add Child」に、メニュー「Add Pattern to node」を表示する(図 32)。このとき、接続可能なパターンを候補として表示する。接続可能かどうかの判断は、現在の子ノードの候補の補完機能と同じ手順で行う。(パターンが単一のツリーであり、ルートノードが、選択されたノードの子ノードとして接続可能かどうかで判断する。)

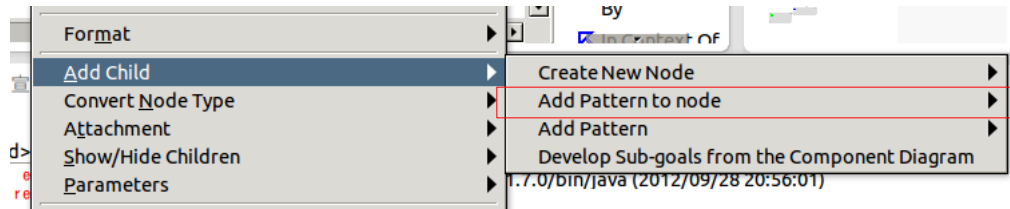


図 32: ノードにパターンを追加するためのメニュー

パターンを選択すると、ノードにパターンを追加する(詳細は 5.4.3 に記載)。

ノードを選択して右クリックし、コンテキストメニューの「Add Child」→「Add Pattern」を選択した際の処理は、現在と同様である。

5.4.3 パターンの追加処理

パターンの追加処理手順の詳細は、以下の通りである。

1. 選択されたパターンが単一のツリーであることを確認する。そうでない場合は、エラーログに下記を出力し、パターン追加処理を終了する。
`パターン名: not a single tree.`
2. パターンのルートノード直下のパラメータノードで定義されているパラメータが、パターン内の各ノードの「Desc Format String」および「Script」属性内で使用されているか確認する。使用されている場合は、現在と同様、パラメータ設定ダイアログを表示し、得られたパラメータ値を使用する。各ダイアログで「キャンセル」ボタンが押された場合は、パターン追加処理を終了する。それ以外で定義されているパラメータについては、設定されているパラメータ値をそのまま使用する。
3. 5.4.1 や 5.4.2 で指定された箇所に、パターンを追加する。5.4.2 の場合は、選択したノードの子ノードとしてパターンを追加する。

6 パラメータ

6.1 概要

現在の D-Case Editor では、パラメータの定義は workspace 全体で共通、パラメータの値はグローバルもしくは各ノードでのみ参照可能となっている。

モジュールおよびパターンの機能を追加するにあたり、パラメータのスコープを、各ノードをルートとするサブツリー単位に設定できるよう、機能を拡張する。そのため、現在はプリファレンスストアにパラメータ定義を格納しているが、本機能拡張では、「パラメータ」ノードの属性に格納する。

また、現在は、ノードにパラメータ値を設定していても、Argument で設定されたグローバルパラメータがある場合、後者が参照される。これを、もっとも近い先祖のパラメータノードで設定されたパラメータ値を参照するようにする。

さらに、d*表現の Argument で設定されたパラメータを、同フォルダ内で共通に参照できるグローバルパラメータとして扱う。

6.2 パラメータ定義および値の設定

指定したパラメータノードでパラメータを定義、もしくは値を設定する手順は、以下の通りである。

パラメータノードを選択して右クリックし、コンテキストメニューの「Parameters」→「Set Parameters」を選択すると表示されるパラメータ設定ダイアログに、パラメータ定義の追加、編集および削除を行うためのボタンを追加する。(図 33)。

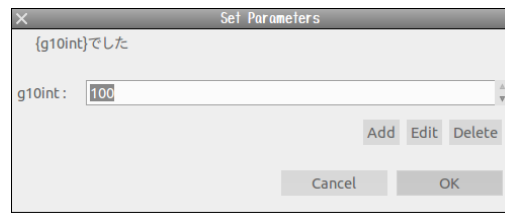


図 33: パラメータ定義および値の設定を行うダイアログ

図 33 のダイアログでは、従来通り、パラメータ値の設定を行うことができる。同時に、パラメータ定義の追加、編集および削除を行うことができる。ちなみに、現在はパラメータ定義の XML ファイルの Import/Export 機能があるが、本機能拡張ではパラメータノードの属性にパラメータ定義が格納されるため、Import/Export ボタンは設置しない。

パラメータ定義を追加した場合、デフォルトのパラメータ値を設定する。

string および raw の場合はサイズが 0 の値、int および double の場合は下限値、enum の場合は最初の項目を設定する。

パラメータ定義を編集した結果、現在の値が範囲外となる場合は、デフォルトのパラメータ値に変更する。

フォルダ内で共通のグローバルパラメータを定義、もしくは値を設定する手順は、以下の通りである。

d*表現の Argument を右クリックし、コンテキストメニューの「Parameters」→「Set Parameters」を選択すると、パラメータノードと同様、パラメータ設定ダイアログを表示する。(図 33)。

ここで定義されたパラメータ、設定されたパラメータ値は、d*表現の属性に格納する。

なお、現在は、Windows メニューの「Preference」→「D-Case Diagram」→「Parameters」でパラメータの定義を行っているが、前述の通り、各ノードで定義を行うようにするため、これを廃止する。

同様に、パラメータの定義および値の設定は、パラメータノードもしくは d*表現の Argument でのみ行われるため、これら以外では、コンテキストメニューから「Parameters」以下を削除する。

6.3 パラメータ値の参照

「Desc Format String」属性内で使用されているパラメータを、以下の順に検索し、設定されたパラメータ値に置換して「Desc」属性に設定する。

1. 自ノードの子ノードにパラメータノードがあり、パラメータ値が設定されている場合、その値に置換する。
2. 親ノードを順にたどり、その子ノードのパラメータノードにパラメータ値が設定されていれば、その値に置換する。親ノードが複数あり、双方に同名のパラメータが設定されている場合は、エラーログに下記メッセージを出力し、置換は行わない。

ノード名: competing パラメータ名.

ただし、複数の先祖ノードに同名のパラメータが設定されている場合でも、最も近いノードが1つに特定できる場合は、最も近いノードのパラメータ値に置換する。

また、現在のダイアグラムを参照する Module ノードがあり(Argument の Userdef011 属性が設定されている)、それが親モジュールと一致する場合は、そのノードを親ノードとして順にたどる。

3. 最上位のダイアグラムのルートノードに達した場合、d*表現の Argument にグローバルパラメータ値が設定されていれば、その値に置換する。
4. 上記によりパラメータ値が見つからなかった場合は、エラーログに下記メッセージを出力する。

ノード名: cannot found パラメータ名.

6.4 パラメータの一覧表示

指定したノードで使用可能なパラメータは、属性入力ダイアログ内に表示する(図 34)。

Name	Value	Type	Node
u1str	U1文字列	string	test2/U_1
s5str	test2のS5よ	string	test2/S_5
g10int	100	int	test2/G_10
attachmentDoub	100.00	double	test1
attachmentRaw	aaaa	raw	test1

図 34: パラメータ一覧を追加した属性入力ダイアログ

パラメータの名前、値、型および設定されているノード(モジュール名/ノード名の形式、ただしグローバルパラメータの場合は空欄)を表形式で表示する。表の行をクリックすると「パラメータ名」を、ダブルクリックすると「{パラメータ名}」をクリップボードにコピーする。

6.5 パラメータ定義および値の格納

パラメータの定義は、ノード属性「Userdef009」に、以下の形式で格納する。

パラメータ名₁, パラメータ名₂, ... ; パラメータ名₁の属性; パラメータ名₂の属性; ...

パラメータ名の使用禁止文字および名称は、現在の D-Case Editor と同じである。

パラメータの属性は、以下の形式で格納する。基本的には、プリファレンスストアに格納されている形式と同等である。

型	形式
string	type=string,min= <u>文字数の下限</u> ,max= <u>文字数の上限</u>
raw	type=raw,min= <u>文字数の下限</u> ,max= <u>文字数の上限</u>
int	type=int,min= <u>下限</u> ,max= <u>上限</u>
double	type=double,min= <u>下限</u> ,max= <u>上限</u> ,digit= <u>小数点以下の桁数</u> ,inc= <u>増減幅</u>
enum	type=enum,items={ <u>項目1</u> , <u>項目2</u> ,...}

パラメータ値は、現在と同様、ノード属性「Userdef007」に、現在の形式で格納する。

このため、すべてのノードにおいて、「Properties」ビューの Userdef009 の属性の表示を、「Parameter Definition」とする。