D-Case Editor マニュアル

26/NOV/2013

改訂履歴

更新日	版	内容
26/NOV/2013	0.9.2	・図を変更(3.3節) ・5 章を追加
7/NOV/2013	0.9.0	新規作成

目次

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」の拡張機能に関連する使用方法を説明するマニュアルです。

1.2 用語の定義

名 称	意 味
ダイアグラム	(D-Case Editor 上で 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。
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

- D-Case 入門
- D-Case Editor ユーザーズマニュアル

  http://www.dependable-os.net/tech/D-CaseEditor/Download_files/DCaseEditorUsersManual100.pdf
- D-Case Editor 機能仕様書

  http://www.dependable-os.net/tech/D-CaseEditor/Download_files/Functional_Description088J.pdf
- D-Case 仕様のリファレンス実装としての D-Case Editor 拡張 機能仕様書
- D-Case 仕様のリファレンス実装としての D-Case Editor 拡張 環境構築手順書

2 モジュール

2.1 概要

D-Case が大規模になり、多数のノードが存在するようになると、理解しづらく、また変更に時間がかかるようになります。

「モジュール」は、D-Case のサブツリーをひとつのまとまりとして扱うためのものです。大きくなった D-Case を、サブツリー単位でモジュールに置き換えることで、D-Case の記述が簡素化され、理解および管理がしやすくなります。

各モジュールは、従来の D-Case と同様、GMF ダイアグラム情報ファイル(サフィックスが dcase_diagram)および GMF モデル情報ファイル(サフィックスが dcase_model)からなります。

D-Case をモジュールに分割した場合、プロジェクト内に、トップのノードを含むモジュールファイルと、分割されたモジュールファイルが存在することになります。

2.2 モジュール化

モジュール化には、大きく分けて 2 つの方法があります。

1 つは、既存の D-Case のサブツリーをモジュール化する方法です。記述した D-Case が大規模になったとき、サブツリーをモジュールに置き換えていくことで、簡素化できます。もう1つは、モジュールを参照するための Module ノードか、モジュール内のノードを参照するための Goal ノード(Away Goal ノードと呼びます)を追加した後、そのノードに参照先を指定する方法です。

既存の D-Case のサブツリーをモジュール化するには、サブツリーのルートノードを右クリックして「モジュールの生成(Create Module)」を選択します(図 1)。

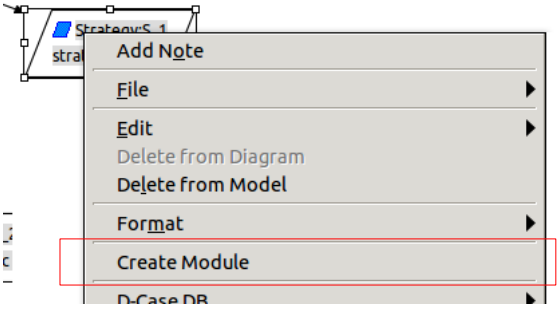

図 1: モジュール化のためのメニュー

モジュール名を入力するためのダイアログ(図 2)が表示されますので、モジュール名を入力して「OK」ボタンを押します。

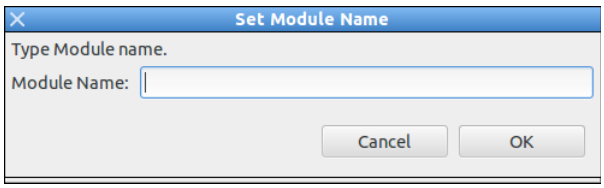

図 2: モジュール名設定ダイアログ

すると、サブツリーがモジュールとして生成され、D-Case 上にあったサブツリーが Module ノードに置き換わります(図 3)。

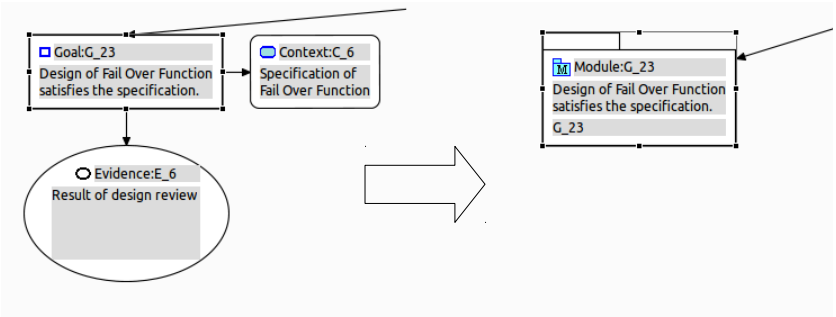

図 3: サブツリーのモジュール化

Module ノードや Goal ノードを追加して、既存のモジュールを参照するには、追加したノードを右クリックして「添付(Attachment)」→「モジュールから選択(Select from Module...)」を選びます(図 4)。

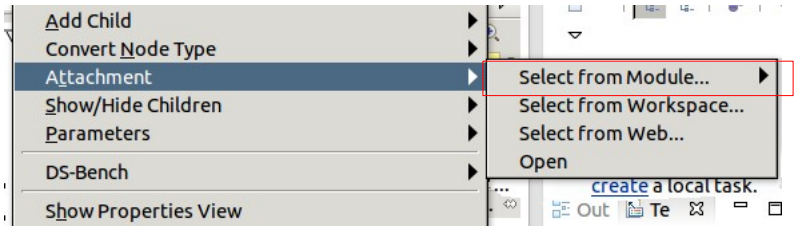

図 4: モジュールを選択するためのメニュー

すると、参照可能なモジュールもしくはノードが表示されます。その中から参照したいものを選びます。

ただし、ノードを参照するには、参照されるノードが「public ノード」である必要があります。参照したいノードを public ノードにするには、そのノードを右クリックして「パブリック/プライベートフラグの設定(Set Public/Private Flag)」の「パブリック(Public)」を選んでください。逆に、参照されたくない場合は、「プライベート(Private)」を選んでください。

2.3 モジュールの展開表示

Module ノードや Away Goal ノードで、参照しているモジュールやノードの内容を知りたい場合は、そのノードを右クリックして「モジュールの表示/非表示(Show/Hide Module)」→「モジュールの表示(Show Module)」を選びます(図 5)。

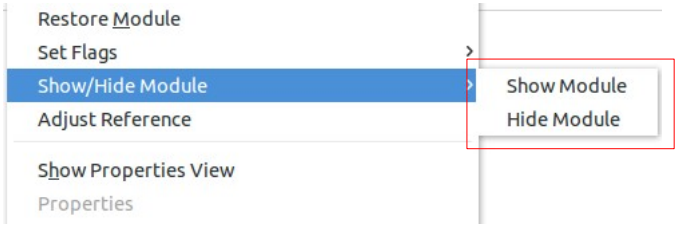

図 5: モジュールを展開するためのメニュー

すると、ノード内に参照先の内容を表示します(図 6)。また、「モジュールの非表示(Hide Module)」を選ぶと、参照先の内容を表示している場合は、もとの表示に戻します。

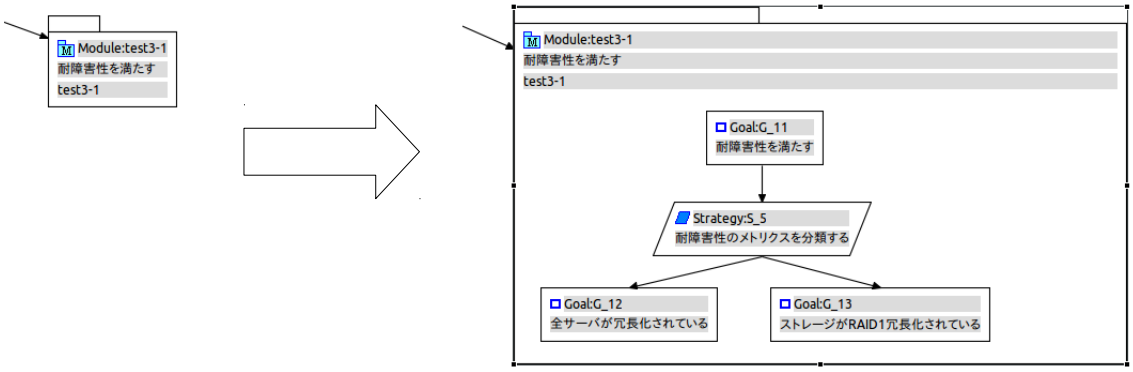

図 6: モジュールの展開表示

2.4 モジュールの解除(展開)

Module ノードをもとのサブツリーに戻すこともできます。

Module ノードを右クリックして「モジュールの展開(Restore Module)」を選びます(図 7)。すると、Module ノードが、参照先のサブツリーの内容に置き換わります。

ただし、参照先のモジュールファイルは削除されません。モジュールファイルを削除するには、次節を参照してください。

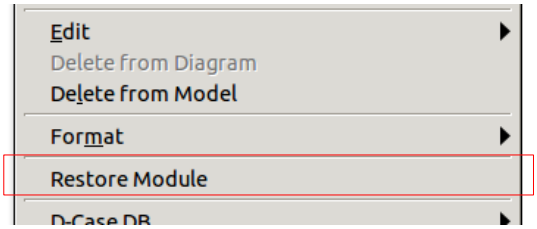

図 7: モジュール解除(展開)のためのメニュー

2.5 モジュールの管理

D-Case をいくつかのモジュールに分割すると、モジュールや public ノードにどのようなものがあるのか、どのノードが参照しているのかなどが理解しづらくなります。その場合は、「Modules ビュー」を使用すると、D-Case のプロジェクト内にあるモジュールおよび public ノードを確認したり、操作することができます。

Modules ビューを表示するには、「Window」メニュー→「Show View」→「Other...」を選び、「D-Case Editor」を展開して、「モジュール(Modules)」を選びます(図 8)。

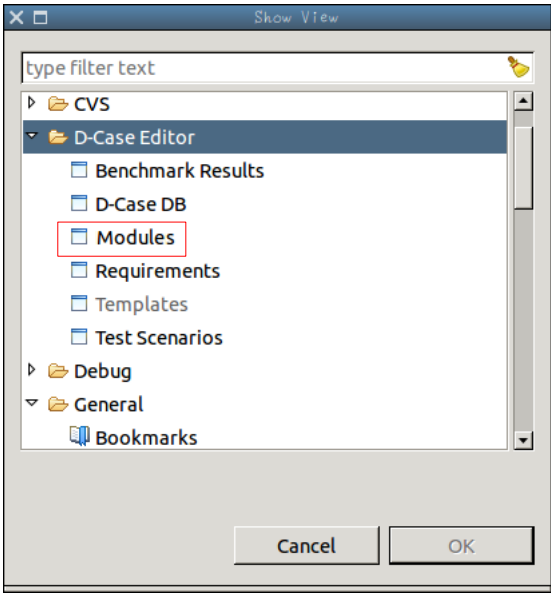

図 8: モジュールビュー表示の選択

すると、Modules ビューが表示されます(図 9)。

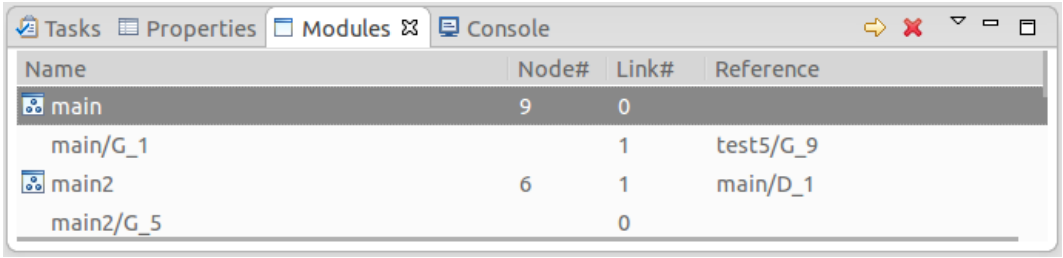

図 9: Modules ビュー

Modules ビューでは、モジュール名および public ノード名、モジュール内のノード数、リンク数、参照元のノード(モジュール名/ノード名)を表形式で表示されます。

モジュールをダブルクリックするか,、モジュールを選んで右上の矢印アイコンを押すと、そのモジュールを開きます。また、モジュールを選んで×アイコンを押すと、どこからも参照されていなければ(リンク数が 0 であれば)、そのモジュールファイルを削除します。

2.6 ノード一覧のテキストファイルへの出力

モジュール内にどのようなノードがあるのか確認したいときなどのために、ノード一覧をテキストファイルに出力することができます。それにはまず、「File」メニュー→「フォーマットの変換(Convert File Type)」→「GMF からテキストへの変換(From GMF to Text)」を選びます(図 10)。

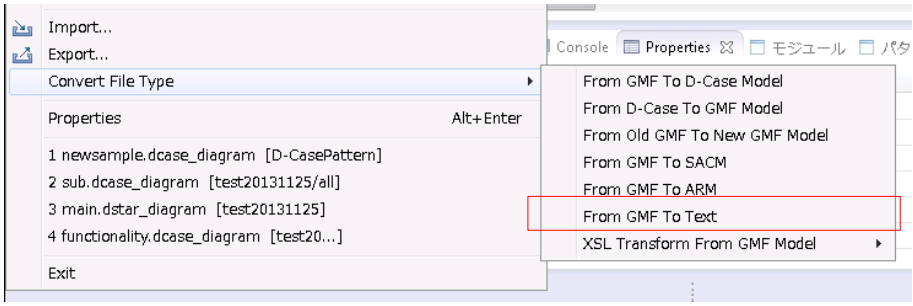

図 10: ノード一覧を出力するためのメニュー

すると、テキストファイルに出力するためのウィザードが表示されます(図 11)。

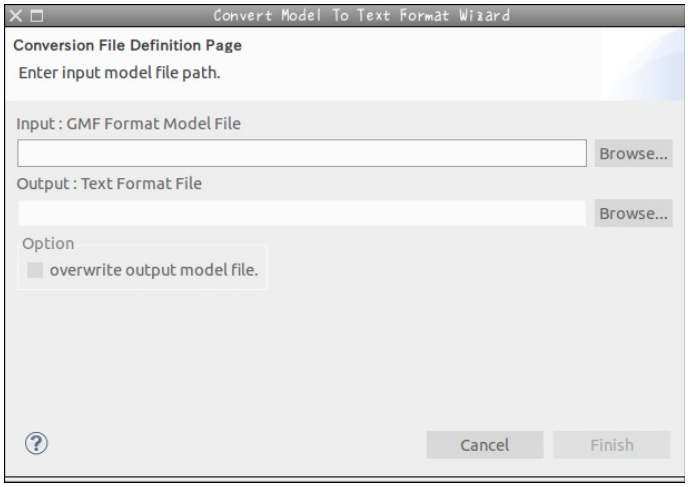

図 11: ノード一覧をテキストファイルに出力するためのウィザード画面

ここで、対象のモジュールである GMF モデル情報ファイルと、出力したいテキストファイルを入力して、「Finish」ボタンを押します。すると、下記のような、テキストファイルが作成されます。

ノード毎に分けて、1 行につき 1 ノードの情報(ノード名、Desc、Attachment(参照先))を出力します。

[Goal]

“G_11","耐障害性を満たす”,""

“G_12","全サーバが冗長化されている”,""

“G_13","ストレージが RAID1 冗長化されている”,""

[Strategy]

“S_5","耐障害性のメトリクスを分類する”,""

[Module]

“M_1","","module1"

…以下略…

3 パターン

3.1 概要

よく使われる可能性のある D-Case を「パターン」として登録しておき、他の D-Case で利用できるようにしておくと、便利です。

D-Case Editor では、「D-CasePattern」プロジェクト内にある D-Case をパターンとして扱います。

3.2 パターンの追加

ダイアグラムの所望の位置にパターンを追加するには、その位置を右クリックして「パターンの追加(Add Pattern)」を選びます(図 12)。すると、パターンの一覧がメニューに表示されるため、その中から追加したいパターンを選びます。パターンを選ぶと、右クリックした位置にパターンを追加します。

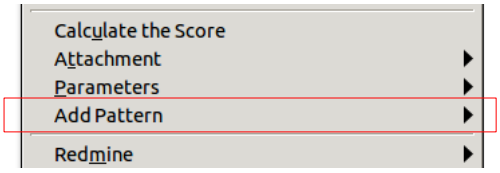

図 12: ダイアグラムにパターンを追加するためのメニュー

ノードの下にパターンを追加するには、ノードを右クリックして「子の追加(Add Child)」→「パターンの追加(Add Pattern to node)」を選びます(図 13)。同様にパターンの一覧がメニューに表示されるため、追加したいパターンを選びます。パターンを選ぶと、そのノードの下にパターンが追加されます。

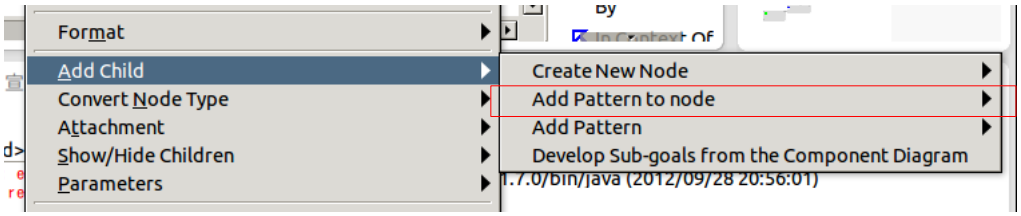

図 13: ノードにパターンを追加するためのメニュー

3.3 Pattern ノードの利用

「Pattern」ノードをパターン内で利用すると、パターンを柔軟に追加することができます。「InContextOf」リンクで Pattern ノードを指すノードをルートとするサブツリーが、処理の対象となります。

Pattern ノードには、「SubType」属性があり、この値によって処理する内容が決まります。取りうる値は、「Parameter」「Loop」「Choice」および「Multiplicity」です。

Parameter は、後述(4章)するパラメータの定義や設定を行うためのものです。

Loop は、パターンを追加する際、対象のサブツリーを、特定のリーフノードに繰り返し連結して追加するためのものです(図 14)。連結対象のリーフノードは、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(AttributeDialog)」の、「LeafNode」属性で指定します。パターン追加の際、Loop が含まれる場合に、繰り返し回数の入力をダイアログ形式で聞いてきます。入力した回数分、サブツリーを繰り返したパターンが追加されます。

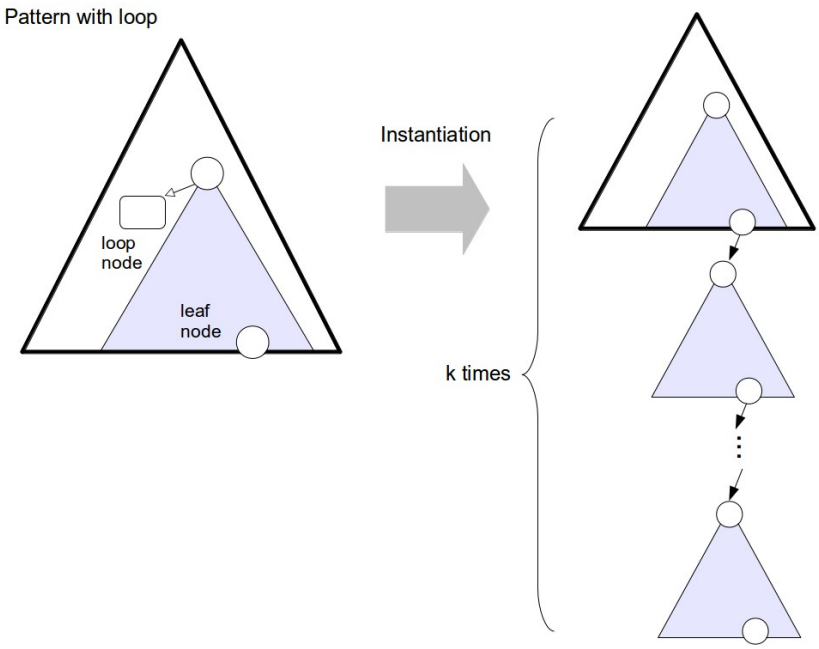

図 14: Loop の構成と処理

Choice は、パターンを追加する際、対象のサブツリーの一部を追加するためのものです(図 15)。対象のサブツリーのルート直下に n 個のノードがあるとき、「i」属性で指定された個数以上、「j」属性で指定された個数以下を追加します。i および j 属性も、属性入力ダイアログで設定します。

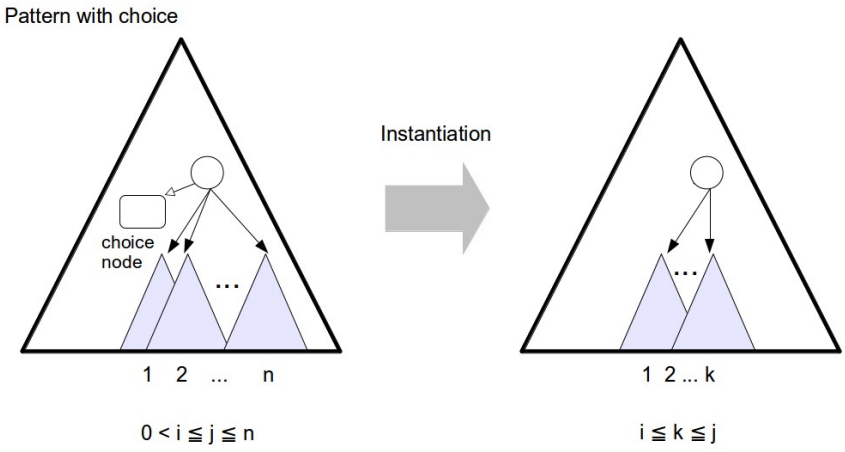

図 15: Choice の構成と処理

Multiplicity は、パターンを追加する際、対象のサブツリーを複数コピーするためのものです(図 16)。対象のサブツリーのルート直下には単一のノードだけがあると想定して、i 属性個以上 j 属性個以下を複製して追加します。

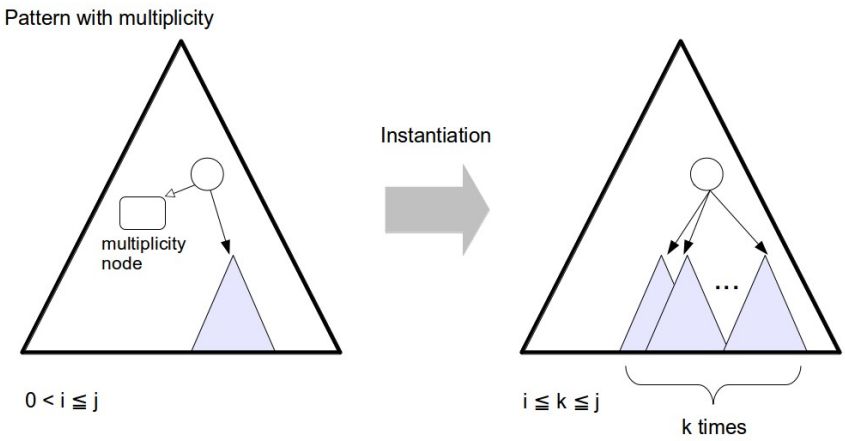

図 16: Multiplicity の構成と処理

4 パラメータ

4.1 概要

モジュールやパターンは、対象のシステムが多少異なっても、同様の構成になることがあります。Desc などの属性にシステム固有の情報を直接記述するのではなく、パラメータを用いて抽象化しておくと、モジュールやパターンを別の D-Case で利用しやすくなります。

パラメータは、Pattern ノードで定義し、値を設定できます。定義・設定されたパラメータは、InContextOf リンクで Pattern ノードを指すノードをルートとするツリー内で使用できます。モジュール化されている場合、親モジュールをさかのぼって、パラメータを参照できます。また、参照可能な同名のパラメータが複数定義されている場合は、近い方のノードのパラメータの値を使用します。

4.2 パラメータの定義と設定

パラメータを定義するには、Pattern ノードを右クリックして「パラメータ(Parameters)」→「パラメータの定義(Define Parameters...)」を選びます(図 17)。

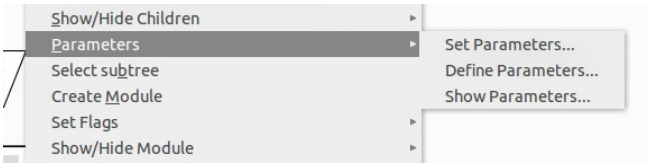

図 17: パラメータ定義のためのメニュー

すると、パラメータの定義を行うためのダイアログが表示されます(図 18)。

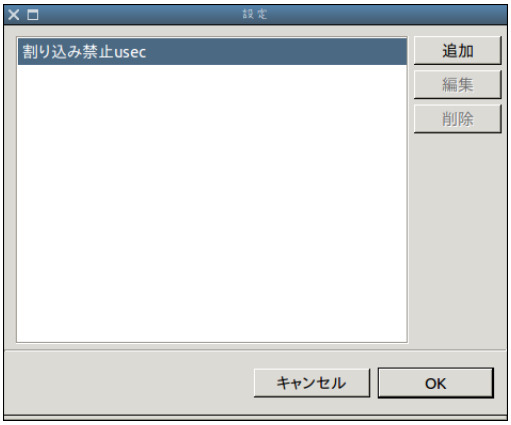

図 18: パラメータ定義を行うダイアログ

「Add」ボタンを押すと、パラメータを追加できます。パラメータ名と型、条件を入力すると、パラメータの定義が追加されます。

パラメータを選んで「Edit」ボタンを押すと、パラメータの定義を変更できます。また、パラメータを選んで「Delete」ボタンを押すと、パラメータが削除されます。

パラメータの値を設定するには、Pattern ノードを右クリックして「パラメータ(Parameters)」→「パラメータの設定(Set Parameters...)」を選びます。すると、パラメータの各値を設定するためのダイアログが表示されます。ここで値を設定・変更できます。

4.3 パラメータの参照

属性入力ダイアログ(AttributeDialog)の「パラメータを使った文章(Desc Format String)」属性は、「文章(Desc)」属性のベースとなるものです。Desc Format String 属性の値のうち、「{パラメータ名}」の形式の文字列が実際のパラメータ値に置き換わり、結果が Desc 属性に設定されます。

ノードで使用できるパラメータは、属性入力ダイアログで確認できます(図 19)。

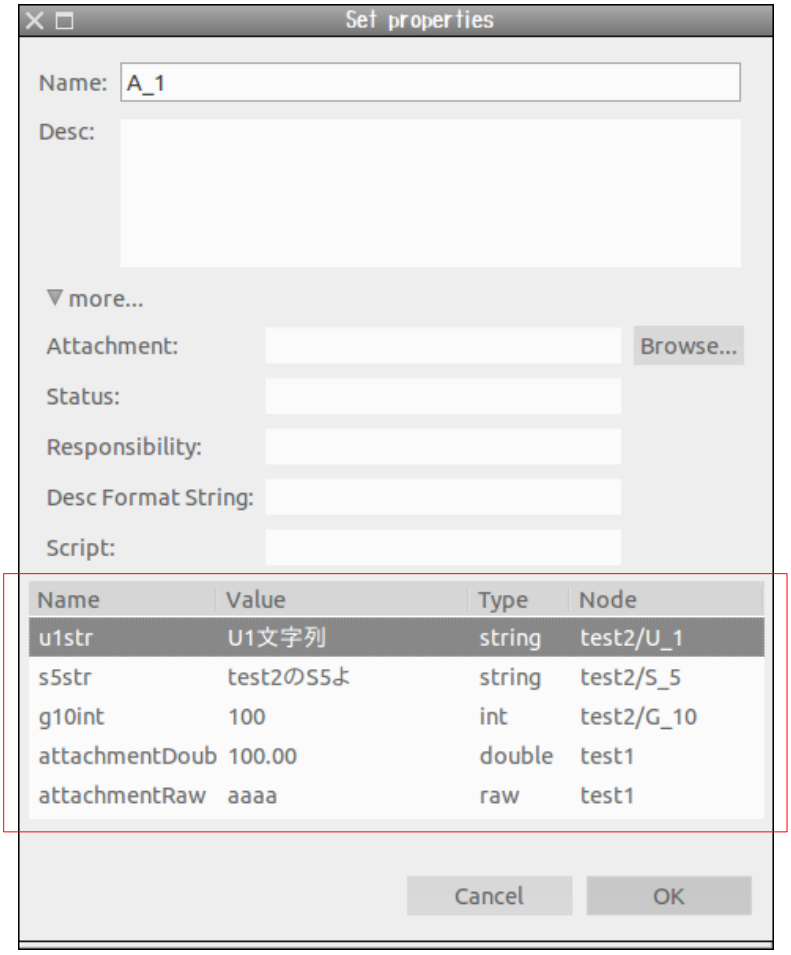

図 19: 属性入力ダイアログ

このダイアログでは、パラメータ名、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)を表形式で表示します。このノードでは参照できないパラメータの情報は、ここには表示されません。

パラメータをクリックすると「パラメータ名」、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します。Desc Format String 属性の入力に使用すると、便利です。

5 その他

5.1 英語表記

D-Case Editor では日本語化を行っているため、日本語環境で Eclipse を起動すれば、D-Case Editor のメニューやメッセージが自動的に日本語で表示されます。

もし、メニュー等を英語にしたい場合は、「-nl en」オプションを指定して Eclipse を起動してください。

```
$ eclipse -nl en
```

ちなみに、「Pleiades」というプラグインをインストールすると、D-Case Editor 以外のメニューなども日本語で表示されるようになります。

http://mergedoc.sourceforge.jp/

5.2 旧バージョンのファイルを扱う

旧バージョン(0.8.15 以前のバージョン)とはスキーマが異なるため、旧バージョンで作成した D-Case(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)を開くことができません。

現バージョンで旧バージョンのファイルを扱うには、変換する必要があります。それにはまず、「File」メニュー→「フォーマットの変換(Convert File Type)」→「旧 GMF から新 GMF への変換(From Old GMF to New GMF Model)」を選びます(図 20)。

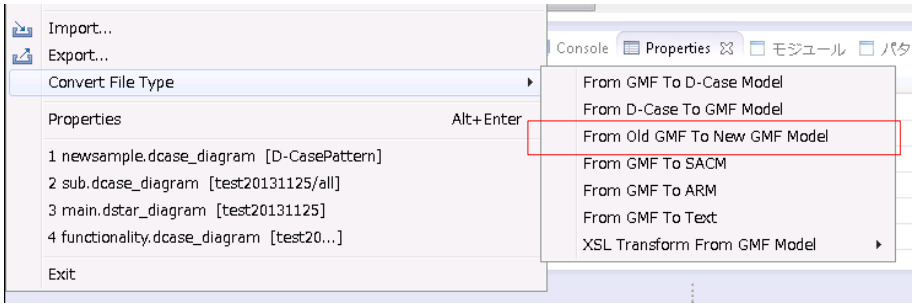

図 20: ノード一覧を出力するためのメニュー

すると、現バージョンに変換するためのウィザードが表示されます(図 21)。

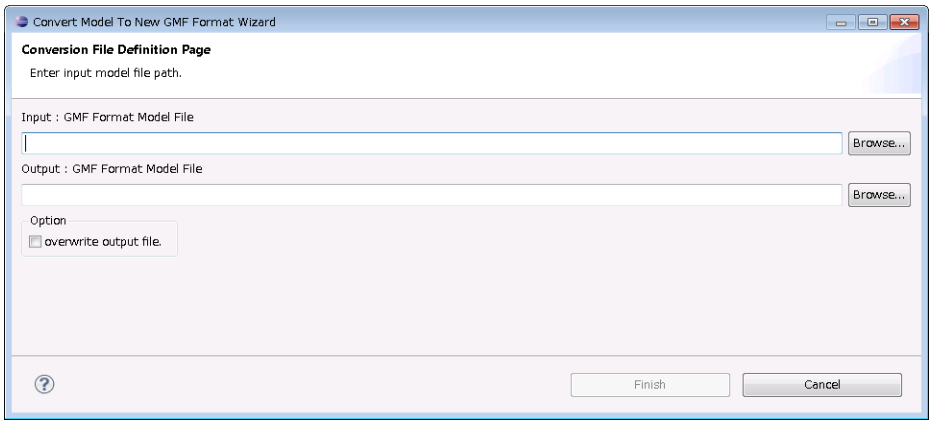

図 21: 現バージョンに変換するためのウィザード画面

ここで、旧バージョンの GMF モデル情報ファイルを選び、出力したいファイル名を入力して、「Finish」ボタンを押します。すると、現バージョンの GMF モデル情報ファイルが作成されます。

次に、GMF モデル情報ファイルを右クリックして、「D-Case ダイアグラムの初期化(Initialize dcase_diagram diagram file)」を選びます(図 22)。

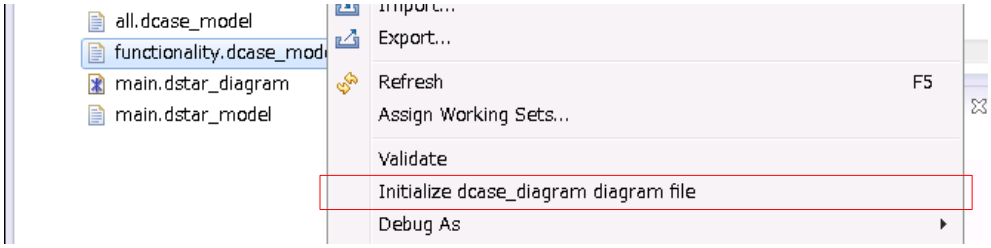

図 22: GMF ダイアグラム情報ファイルを生成するためのメニュー

すると、GMF ダイアグラム情報ファイルが生成されて、ダイアグラムを開きます。

同プロジェクト内に旧バージョンの D-Case が含まれていると、Modules ビューや d*などで問題になります。ですので、旧バージョンと新バージョンは同じプロジェクトに含めず、分けることをお勧めします。