

D-Case EDITOR

ユーザーズマニュアル



Ver. 1.0.1 日本語版

改訂履歴

更新日	版	内容	担当
1/MAY/2015	1.0.1	・デザイン、レイアウト変更、Typo 修正 ・参考文献変更 ・2.1、3.3の項目を修正	石垣@DIMENSIONS
12/APR/2015	1.0.0	・1.3の項目を追加・修正 ・2章を追加 ・3章を追加 ・7.3を追加	E∃田@AXE
26/NOV/2013	0.9.2	・図を変更(16節) ・21章を追加	臼田@AXE
7/NOV/2013	0.9.0	新規作成	臼田@AXE

目次

1	は	じめに	4
	1.1	概要	4
	1.2	用語の定義	4
	1.3	関連文書	4
2	イン	ンストール手順	5
	2.1	環境	5
	2.2	インストール手順	5
	Ec	lipse のインストール	5
	D-	Case Editor プラグインのインストール	5
	パ	ターンプロジェクトのインポート	6
3	基	本的な使い方	7
	3.1	プロジェクトの作成	7
	3.2	D-Case 文書の作成	7
	3.3	ノード・リンクの作成	7
	3.4	自動配列	9
4	モ	ジュール	. 10
	4.1	概要	. 10
	4.2	モジュール化	. 10
	4.3	モジュールの展開表示	. 11
	4.4	モジュールの解除(展開)	.12
	4.5	モジュールの管理	.13
	4.6	ノードー覧のテキストファイルへの出力	.14
5	パ	ターン	.16
	5.1	概要	.16
	5.2	パターンの追加	.16
	5.3	Pattern ノードの利用	.16
6	١٩	ラメータ	. 19
	6.1	概要	. 19
	6.2	パラメータの定義と設定	. 19
	6.3	パラメータの参照	.20
7	そ	の他	.21
	7.1	英語表記	.21
	7.2	旧バージョンのファイルを扱う	.21
	7.3	SACM への 変換	.22

1 <u>はじめに</u>

1.1 概要

本書は、型チェック機能などを持つ、アシュアランスケースのエディタである「D-Case Editor」の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

・ 松野裕, 山本修一郎, 高井利憲, D-Case 入門, ダイテックホールディング, ISBN 978-4-86293-079-8

- ・ D-Case ウェブサイト (http://www.dcase.jp/)
- The Eclipse Foundation (http://www.eclipse.org/)
- JRE java.com (http://java.com)
- · 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, ISBN 978-4-8399-2972-5

2 <u>インストール手順</u>

2.1 環境

本マニュアルでは、下記の環境で D-Case Editorを構築します。

※GMF Runtime および OCL Classic SDK は、D-Case Editor のインストールの際、自動的にイン ストールされます。

- Microsoft® Windows® 7 Professional SP1
- Oracle Java SE Runtime Environment 7 Update 75
- Eclipse IDE for Java Developers (4.4.2 / Luna 32bit)
- Graphical Modeling Framework(GMF) Runtime (1.8.1)
- OCL Classic SDK (5.0.2)
- ・ パターンプロジェクト (D-CasePattern.zip)
- ・ ZIP ファイルを解凍できるアーカイブユーティリティ(+Lhaca など)

2.2 インストール手順

はじめに、JRE(Java Runtime Environment)がインストールされた Windows PC を準備してください。 次に、以下の手順で D-Case Editor をインストールします。

Eclipse のインストール

1. Eclipse のサイトにアクセスし、「Downloads」をクリックして、「Eclipse IDE for Java Developers」の Windows 32bit 用のアーカイブファイルをダウンロードします。

2. ダウンロードしたアーカイブファイルを、アーカイブユーティリティを使用して、任意の場所 で展開します。

3. 「eclipse.exe」を起動します。

D-Case Editor プラグインのインストール

- 1. Eclipse を起動します。
- 2. 「Help」メニューの「Install New Software...」を選択します。
- 3. Work with に「http://dimensions-japan.org/dcase/eclipse/」を指定します。
- 4. 「D-Case Editor」にチェックして、インストールを行います。

もし表示されない場合は、「Group items by category」のチェックを外してください。

5. 以下のダイアログが表示される場合は、「OK」ボタンをクリックしてください。

0	Warning: You are installing software that contains unsigned content. Th authenticity or validity of this software cannot be established. Do you w to continue with the installation?	në Vant
	OK Detaile as	

パターンプロジェクトのインポート

- 1. Eclipse を起動します。
- 2. 「File」メニューの「Import…」を選択します。
- 3. 「General」を開き、「Existing Projects into Workspace」を選択し、「Next」ボタンをクリックします。
- 4. 「Select archive file」にチェックし、「Browse…」ボタンをクリックして、パターンプロジェクトの アーカイブファイル「D-CasePattern.zip」を選択します。
- 5. 「Copy projects into workspace」にチェックして、「Finish」ボタンをクリックする。

3 <u>基本的な使い方</u>

3.1 プロジェクトの作成

D-Case 文書を格納するためのプロジェクトを作成します。

- 1. 「File」メニューの「New」→「Project」を選択します。
- 2. 「General」を開き、「Project」を選択して、「Next」ボタンをクリックします。
- 3. 「Project name」にプロジェクト名を入力して、「Finish」ボタンをクリックします。

3.2 D-Case 文書の作成

D-Case 文書を作成します。

1. 3.1で作成したプロジェクトを右クリックして「New」→「Other...」を選択します。

2. 「D-Case Editor」を開き、「D-Case Diagram」を選択して、「Next」ボタンをクリックします。

3. 「File name」にファイル名を入力して、「Next」ボタンをクリックします。このとき、拡張子「.dcase_diagram」は変更しないようにしてください。

4. 「Finish」ボタンをクリックします。

3.3 ノード・リンクの作成

3.2で文書を作成すると、エディタが表示されます(図1)。

800 Plug-in Development - dca	se_sample/test.dcase_dlagram - Eclipse	
	- 4 ・ 単 6 ・ 29 2 - 1 ・ 21 ・ 21 ・ 4 ・ 4 ・ 4 ・	
Abyssinica SIL - 9	▼ B / A · A · / · → · 顎 淡・鸣・巻・ 舀 / × 日・ 100% - ▼	
	Quick Access 🔡 😫 🛛 🚭 Plug-in Develop	ment
😫 Package 😫 🌫 Plug-ins 📟 🥅	∦ test.dcase_diagram ¤	
□ % 5 ▼		₽
D-CaseEditorUpdateSite		* *
D Gase_sample		. 0
P 🙀 > net.dependableos.dcase 239	N O証拠	
P 11 > net.dependableos.dcase.diag	/ 戦略	
P 13 > net.dependableos.dcase.diag	□前提	
P 13 > net.dependableos.dcase.diag	T Difference	
P 13 > net.dependableos.dcase.edit	(m = - 2	*** **
P i net.dependableos.dcase.featur	ロバターン	/
	© @ アクショ	v
	室 外部接近	統
Be Outline # I Task List	> 1>2	0
1 H 1 V	∞ Kuppor	rted
	By	
	· Of	text
	~	性
	? Problems Target Platform State Console モジュール Properties	
	Properties are not available	
	+C.	

図 1: エディタの画面

ノードを追加するには、画面右側の「Palette」リストからノードを選択し、画面中央のキャンパスを クリックします。クリックした場所にノードが追加されます。 ノード間の関連を示すリンクを追加するには、同様に「Palette」リストからリンクを選択し、ノード間 をドラッグします。あるいは、ノードを右クリックして「子の追加」→「ノードの追加」→ノードの種類 を選択すると、選択したノードとリンクが追加されます(図2)。

test.dcase_diagram		•
<u></u>)	6810121416	tte
	Add Note	/= ř
Ň	File	•
	Edit Delete from Diagram Delete from Model	
t [Format	• • • •
 証拠 ブードの追加 戦略 パッ・フの追 前提 正当化 仮定 未達成 モジュール 契約 モニタ パターン アクション Antesta 	 子の追加 予つでの推測を変更 添付 子の表示/非表示 サブツリーの選択 モジュールの作成 パブリック/プライペートフラグの設定 モジュールの表示/非表示 外部コマンド リフレッシュ d*表現の作成/更新 	▶
Userdef002 Target Platform Userdef003	State Show Properties View Properties	_
Console	Remove from Context Shift+Ctrl+Alt+Do	vn 🗋

図 2: 子の追加

ノードの属性を設定するには、ノードをダブルクリックすると表示される「属性ダイアログ」で行うか (図3)、

				· · · 参照…
				参照
				Ξ
				
				参照
1 ÷				_
0				
			ŀ	+
	型	ノード		
			1 ÷ 0 型 ノード Cancel	1 ÷ () 型 ノード Cancel

図 3: 属性ダイアログ

「Properties」ビューで行います(図4)。「Properties」ビューは、「Window」メニューの「Show View」で「Properties」を選択すると、表示されます。(「Other」→「General」に入っている場合もあります)

Property	Value
パラメータを使った文章	71
パラメータ定義	12 12
フラグ	12
プロジェクト	12
メッセージ	12
参照元	12
重み	年 1
44:00	735

図 4: Properties ビュー

3.4 自動配列

作成した D-Case 文書を自動配列できます。「Diagram」メニューの「Arrange」→「All」を選択すると、 垂直方向に自動配列されます(図5)。



図 5: D-Case 文書の自動配列

「D-Case」メニューの「整列」→「水平方向」を選択すると、水平方向に配列することもできます。

4 モジュール

4.1 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。

4.2 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6)。



図 6: 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7)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。 D-Case Editor ユーザーズマニュアル

× Set M	odule Name	
Type Module name. Module Name:		
	Cancel	ОК

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8)。



図 8: 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9)。

Add Child Convert Node Type	
A <u>t</u> tachment	Select from Module
<u>S</u> how/Hide Children <u>P</u> arameters	 Select from Workspace Select from Web
DS-Bench	Open
Show Properties View	·····································

図 9: 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。

4.3 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10)。



図 10: 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11)。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 11: モジュールの展開表示

4.4 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12)。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。

Edit	•
Delete from Diagram	
De <u>l</u> ete from Model	
For <u>m</u> at)
Restore Module	
D-Case DB	

図 12: モジュール解除(展開)のためのメニュー

4.5 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13)。

type filter tex		
	t	\$
D 🗁 CVS		-
🝷 😂 D-Case B	Editor	
Bench	nmark Results	
🗖 D-Cas	e DB	_
🗖 Modu	ıles	
🗖 Requi	irements	
🗖 Temp	lates	
Test S	cenarios	
👂 🗁 Debug		
🗢 😂 General		
Book	marks	

図 13: 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14)。

🖉 Tasks 🖾 Properties 🗖 Modules 🛿	Console			🗢 🗙	▽ □	
Name	Node#	Link#	Reference			
🚨 main	9	0				
main/G_1		1	test5/G_9			
🚨 main2	6	1	main/D_1			
main2/G_5		0				

図 14: 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。

4.6 ノードー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、ノードー覧をテキストフ 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15)。

	Import Export	Console Properties 23 日モジュール ロバ		
	Convert File Type		From GMF To D-Case Model	
	Properties	Alt+Enter	From D-Case To GMF Model	
1 newsample.dcase_diagram [D-CasePati 2 sub.dcase_diagram [test20131125/all] 3 main.dstar_diagram [test20131125] 4 functionality.dcase_diagram [test20]	1 newsample.dcase_diagram [D-CasePattern] 2 sub.dcase_diagram [test20131125/all]		From GMF To SACM From GMF To ARM	
	3 main.dstar_diagram [test20131125]		From GMF To Text	
	4 functionality.dcase_diagram [test20]		XSL Transform From GMF Model	
	Exit	xit		

図 15: ノードー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16)。

×□	Convert Model To Text Format	Wizard
Conversion File Defir Enter input model fi	nition Page le path.	
Input : GMF Format	Model File	
		Browse
Output : Text Forma	t File	
		Browse
Option	ıt model file.	
(?)		Cancel Finish

図 16: ノードー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...

5 パターン

5.1 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。

5.2 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17)。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 17: 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18)。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。

	For <u>m</u> at	By Concenter of
-	Add Child	Create New Node
-	Convert Node Type	Add Pattern to node
	A <u>t</u> tachment	Add Pattern
d>	Show/Hide Children	Develop Sub-goals from the Component Diagram
ere	<u>P</u> arameters	1.7.0/bin/java (2012/09/28 20:56:01)

図 18: ノードにパターンを追加するためのメニュー

5.3 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(6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 19)。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 19: 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 20)。 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。

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 21)。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 21: Multiplicity の構成と処理

6 パラメータ

6.1 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。

6.2 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22)。

	Show/Hide Children	٠	
	<u>P</u> arameters	Þ	Set Parameters
Γ	Select su <u>b</u> tree		DefineParameters
(Create <u>M</u> odule		Show Parameters
	Set Flags	+	
-	Show/Hide Module	Þ	

図 22: 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23)。

	設定	
割り込み禁止usec		追加
		編集
		削除
	キャンセル	ОК

図 23: 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。

6.3 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 24)。

X 🗆	Set p	roperties	
Name: A_1			
Desc:			
▼more			
Attachment:			Browse
Status:			
Responsibility:			
Desc Format Str	ing:		
Script:			
Name	Value	Туре	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	6666	raw	test1
		Cancel	OK

図 24: 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。

7 <u>その他</u>

7.1 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/

7.2 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25)。

Import Export	Console Properties 🕄 🗖 モジュール 🗖 パ	
Convert File Type		From GMF To D-Case Model
Properties	Alt+Enter	From D-Case To GMF Model
	From Old GMF To New GMF Model	
1 newsample.dcase_diagram [D-CasePattern]	From GMF To SACM	
2 sub.dcase_diagram [test20131125/all]	From GMF To ARM	
3 main.dstar_diagram [test20131125]		From GMF To Text
4 functionality.dcase_diagram [test20]		XSL Transform From GMF Model
Exit		

図 25:「フォーマットの変換」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26)。

D-Case Editor ユーザーズマニュアル

Convert Model To New GMF Format Wizard		
Conversion File Definition Page		
Enter input model file path.		
Input : GMF Format Model File		
1		Browse
Output : GMF Format Model File		
		Browse
Option		
overwrite output file.		
(?)	Finish	Cancel

図 26: 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27)。

 all.dcase_model functionality.dcase_model main.dstar_diagram main.dstar_model 	ы С	Expot		-
	Co	Refresh Assign Working Sets	F5	23
	ſ	Validate		
		Initialize dcase_diagram diagram file		
	•	Debug As	•	

図 27: 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。

7.3 SACM への変換

D-Case Editor では、D-Case 文書を「SACM(Structured Assurance Case Metamodel)」に変換でき ます。図25と同様、「File」メニューの「フォーマットの変換」→「GMF から SACM への変換」を選択 します。

パラメータなどに対応した「拡張 SACM」へ変換するには、同様に「File」メニューの「フォーマットの変換」→「GMF から拡張 SACM への変換」を選択します。