

CASEデータ交換言語STLによる複数方法論の取扱い

3 U - 3

湯原 義彦 岡山 敬 松本 憲幸

(株)東芝 府中工場

1. はじめに

多様なCASEツールの普及にともない、CASEツール間での情報交換が要求されるようになり、情報交換形式の標準化作業が進められている。IEEEの試用DraftであるSTL (Semantic Transfer Language)[1]は、構造化分析などの上流工程で作成される各種ダイアグラムに関する情報交換形式案である。実際にSTLを利用して複数のCASEツール間での情報交換を行う場合には、CASEツールごとにサポートしている方法論が異なるため、STLの複数の設計方法論への適用性が問題になる。我々は、複数の方法論にSTLを適用する際に発生する問題について検討し、情報交換ツールの試作を行った。ここでは、その実現方法及び有効性について報告する。

ConnectionPath	pathA	
has label		"LABEL";
has description		NULL;
is connectiontype		data;
:		:
:		:
connects from action		processA;
connects to action		processB;
carries dataitem		dataA;
:		:

図2-2 STLによるアローシンボル記述例

2. 問題点

2.1 ダイアグラムシンボルの種類の違い

同種のダイアグラムでも、方法論によって記述するシンボルに相違がある。例えば、リアルタイム用途の方法論[2][3]にはコントロールを意味するシンボルが存在し、さらにWard法では処理プロセスがデータ処理とコントロール処理に分離される。このような方法論ごとに異なるシンボルの種類に対して、STLのConceptの種類を拡張を行うかConceptの属性の拡張を行うかによって、拡張されたSTLの形式に相違が生じる。また、シンボルの種類の違いは、ある設計図を他の方法論に移した場合の情報交換の可否や情報の縮退・欠落などの問題を生じる。

2.2 グラフィック情報の定義

STLの試用Draftでは、設計図の物理的な図形表現上でシンボルの形状を識別する is graphicsymboltype 属性の値として実際にどのような種類のグラフィックシンボルを用意するかは決定されていない。情報交換を行う場合には、シンボルの形状が処理系依存にならないように属性値を決め、かつ方法論間で生じる同種異形のシンボルを表現の取扱いを決定する必要がある。(図2-1)

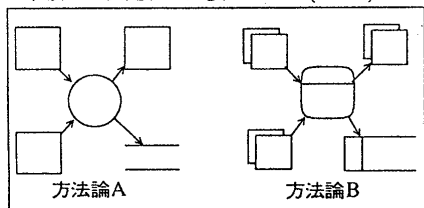


図2-1 シンボル形状の差異

2.3 シンボル結合の方向性の問題

STLではプロセス(Action)間を流れる情報のフローは、あるプロセスから別のプロセスへ結合したConnectionPathによって表現され、暗に方向性が想定されている(図2-2)。一方、方法論によっては双方向の情報フローの記述が存在するが、STLの仕様範囲内では1つのConnectionPathによって双方向性をもったフローを表現することができないので、その表現方法を決定しなければならない。

3. 解決方法

3.1 シンボルの種類の相違の取扱い

方法論間における設計情報シンボル相違は、記述された設計図に次の2つの異なった解釈を生じ得る。

- (1) 設計情報が排他されているという解釈
ある方法論に基づいた設計図で使用されていないシンボル(例えばコントロールフロー)は、意識的に排除されて設計されているという解釈である。この種の設計図は、Hatley法を習得した後に、積極的にYourdon法[4]を採用して設計を行う場合等に作成され得る。
 - (2) 設計情報が縮退しているという解釈
ある手法で設計を行う場合、他の設計方法論における設計情報との対応をまったく意識しないで設計を行っているという解釈である。この種の設計図は、ただ1つの設計手法を習得して設計を行う場合に作成され得る。この場合、方法論ごとに情報の分解能が異なるため、設計情報は方法論によって縮退されていると解釈される。
- この2つの解釈は、設計者に依存したものであり、処理系が自動的に判別することができないため、複数の方法論間での情報交換をサポートする場合、それぞれの解釈に基づいた変換モードを用意する必要がある。各変換モードで、設計図は図3-1、図3-2のように変換される。

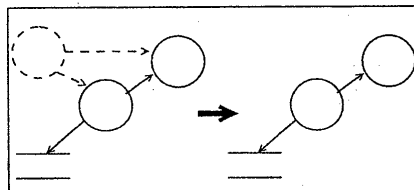


図3-1 排他的解釈モードでの変換

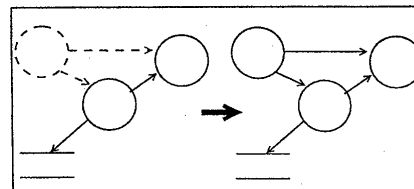


図3-2 縮退解釈モードでの変換

3.2 グラフィック情報の取扱い方法

STLにおいて、物理的なシンボルの形状はグラフィックシンボルの graphicsymboltype 属性値で識別され、その値は1つの文字列で表されることが要求される。

一般には Actionや ConnectionPath等の各種 Concept やその属性に依存してシンボルの形状は異なり、同種の属性をもったシンボルでも、方法論に依存して物理的な形状が異なる。故に、シンボルの形状を決定するために最低限必要な情報は、以下の2つの情報である。

- どの方法論でダイアグラムを表現するか。
- シンボルの Concept種別とその属性。

前者は、1つのダイアグラムに関して唯一指定されるべきものであるから、STLの identification-sentence の属性節として記述するような拡張が自然である。

後者は、graphicsymboltypeの属性値に対応する情報であり、最も単純な属性値の決め方は、グラフィックシンボルに対応する sentenceの Concept種別およびその属性値をそのまま graphicsymboltypeの属性値として採用することである。例えば、データ処理プロセスに対応するグラフィックシンボルの種別としては、コンセプト名称 "Action"、actiontype属性 "internal"、transform purpose属性 "data" の3つによって指定する。(図3-3、図3-4)

Action	processA
has label	"LABEL";
has description	NULL;
is actiontype	internal;
:	:
:	:
has transform purpose	data;
:	:
:	:

図3-3 STLによるシンボルの論理的定義

GraphicSymbol	graphicA
has label	"LABEL";
is graphicsymboltype	"Action-internal-data";
has location	X, Y;
has symbolsize	NULL;
has symbolscale	1, 1;
:	:
:	:

図3-4 グラフィックシンボルの定義

グラフィックシンボルの型をこの方法で指定することは、次の2つの利点がある。

- graphicsymboltype の属性値が設計方法論に依存しない。例えば、Gane法^[5]と Yourdon法の間で処理プロセスのグラフィック属性が一致する等。
- graphicsymboltype の属性値が処理系に依存しない。例えば、ある処理系がプロセスの形状を「丸」と呼ぶか「円」と呼ぶか等に依存しない。

3.3 双方向フローの取扱い

双方向フローに関しては1つの ConnectionPathの属性で表す方法と、2方向の2つの ConnectionPathで表す方法が考えられる。ここでは、1つの双方向フローが論理的には2つの方向性を持ったフローの簡略化された表現である事実を処理系に記憶することなく表現するために、論理情報に関しては後者、グラフィック情報としては前者をとっている。(図3-4)

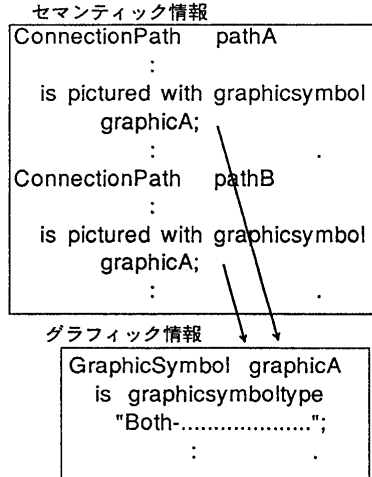


図3-4 拡張STLによる双方向アローの表現

4. おわりに

コントロールの取扱いやシンボルの相連など、複数の方法論に対応して実現した STL変換処理の概念と効果について説明した。設計方法論間での変換は、実際に実行すると排他的な解釈や縮退した解釈のいずれの変換モードでも Hatley法から Yourdon法への変換等によって情報の欠落を生じることになる。ここで説明した STLの解釈は、方法論間での変換をサポートすることよりも、複数の方法論で STLを使用する場合に最も統一的な表現・解釈を与えるように STLの拡張を行うことを目的としている。

ここで説明したダイアグラム情報の表現の違いの他にも、方法論によって用意される設計図の種類の違いや、オブジェクト指向分析手法との対応関係など、さらに検討を行って行く予定である。

[参考文献]

- [1] A Standard Reference Model for Computing System Tool Interconnections, IEEE, Inc., 1991
- [2] Derek J. Hatley, Strategies for Real-Time System Specification, Dorset House Publishing Co., Inc., 1986
- [3] Ward, Paul T., and Stephen J. Mellor. Structured Development for Real-Time Systems. Englewood Cliffs, N. J.: Prentice-Hall, 1985
- [4] Edward Yourdon, Managing the Structured Techniques, Prentice-Hall, Inc., 1986
- [5] C. Gane and T. Sarson, Structured Systems Analysis: Tools and Techniques, IST, Inc., 1977