

5J-01 木構造図用データ交換言語 DXL(1) - 言語仕様 -

5J-1

松尾尚典、松村一夫†

株式会社東芝 研究開発センター システム・ソフトウェア生産技術研究所‡

忠海均、長野宏宣§

日本電信電話株式会社 ソフトウェア研究所¶

1 はじめに

DXL(Diagram eXchange Language for tree structured charts)は異なる種類の木構造図 CASE ツールで作成したデータを、ツール間で相互に交換するために開発されたデータ交換言語である。DXLは、日本規格協会 INSTAC STD-WG3分科会(1988-91)で規格案(第1版)が提案され、その後情報処理学会情報規格調査会 SC7/WG1小委員会(1993年10月からはWG11)において実用化のための評価実験、改良を行い、国際標準化項目の一つとして1992年の国際会議に文書提案し各国との議論を開始している。

本稿ではこのDXL規格案の目的、そしてDXLの設計方針と構文規則の概要について述べる。更にDXLを使ったデータ交換実験について触れ、その結果を基にしたDXL規格改良案について述べる。

2 DXL 規格案作成の背景と目的

現在、木構造図、及び対応する木構造ツールの標準化がなされていないため、複数社による大規模ソフトウェア開発等では各社が異なる木構造図を使用していることにより種々の弊害が発生する。

例えば納入するドキュメントの記法を統一するため、一つの木構造図を開発に用いるとすると、これにより他の木構造図を利用してきた開発者は新しく別の木構造図を学ばなければならない。更に自社のCASEツールを使うことが出来なくなる。

この様な弊害に対処するために、既存の木構造図の一つを標準としたり、新たな標準木構造図を制定することが考えられる。しかしこれらの手段は新たな技術者教育、木構造図設計図資産の再構成などの多大な費用を必要とする。そこで現実解として、既存の木構造図を相互交換可能とする標準となる交換形式が求められた。

DXLは図1の様にそれを介することで、各社木構造図ツールデータを交換し、設計情報の流通、再利用を図ることを目的とする。

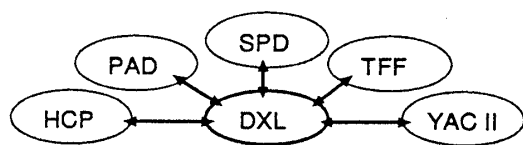


図1: DXLによるデータ交換

*Diagram Exchange Language for Tree-structured Charts DXL(1) - Syntax -

†MATSUO Hisanori, Kazuo MATSUMURA

‡Systems&Software Engineering Lab., R&D Center, Toshiba Corp.

§Hitoshi TADAUMI, Hironobu NAGANO

¶Software Engineering Lab., NTT Corp.

3 DXL 規格案

3.1 適用範囲

DXLによる交換の適用範囲となる木構造図は、ISO8631 (JISX0128)に準拠した図であり、DSD、HCP、PAD、PSD、R-chart、SPD、TFF、及びYAC IIである。ISO5807 (JISX0121)のプログラム流れ図も構造化された記述であれば適用範囲内と考える。

表現出来る記号は、ISO8631 (JISX0128)のプログラム構成要素に対応する記号と、上記木構造図が共通に用いている記号である。

取り扱うことの出来る情報は、木構造図が表現するモジュール内のアルゴリズムに関する情報である。データやグラフィック表現に関する情報、木構造図の管理情報等は現在は扱わない。

3.2 設計方針

DXL規格案の構文規則は、以下に示す設計方針に基づき作成された。

- DXLはファイルや通信バスを介してデータ転送が容易で、且つ可読性があるように、テキスト形式の言語とする。
- DXLは下流CASEツールを対象とする。上流CASEツールを対象にした、他の国際的な標準(SC7/WG11で検討中)と整合性を取る。
- 現状で普及しているツールを生かすため、基本的には各木構造図/ツールの和を取る方向でDXLを定める。DXLを読み込む側のツールは自分の持つ構造と合致しない情報は捨てるか、又は何らかの交換ルールを決めて取り込むこととする。
- コード生成は直接の目的とはせず、設計レベルでのアルゴリズムの表現に限ることを原則とする。

3.3 データ交換実験

国内の木構造図ツールを開発している主要な企業が参加して、基礎的な相互交換実験を行い、実用化に向けて評価、問題点を明らかにした[2]。

4 DXL 規格改良の方針

4.1 規格案の問題点

データ交換実験の過程及び結果から以下の問題点が明らかになった。

木構造図の交換上の問題 DXL要素と木構造図の記号要素が1対1に対応しないことにより、交換誤りが発生したり、交換方法が一意に定まらない等の問題が発生している。

DXLの仕様上の問題 意味不明瞭、曖昧な構文、必要性の不明な構文があった。例えば繰り返しを行う構文は5種類あり、繰り返し条件の意味の区別が曖昧である。

4.2 改良案の要点

データ交換実験結果と、交換ツール作成者及び実験担当者の意見をふまえ、SC7/WG11でDXL改良案をまとめた。改良案の要点は次の2点である。

コンフォーマンスレベル(CF)の導入 木構造図の詳細度に交換レベルを設定し、その範囲内での交換を保証するという考えを導入する。

不明瞭 / 曖昧 / 不要な構文規則の改善 例えば繰り返し条件を整理し、CF-0では1種類、CF-1では3種類にする。

4.3 コンフォーマンスレベル導入の考え方

木構造図 CASE ツール間でデータ交換を実施するに当たっては、幾つかの利用法が想定される。例えば、

- 異なる木構造図を採用している複数社で共同開発する場合
- 既存システムを再利用する場合

等が考えられる。この様な利用法の中には、情報を落さない完全な交換が必要な場合もあれば、ある程度の情報落ちがあってもエラーや警告を出し、手作業でそれを修正すれば運用に問題がない場合等、様々である。

データ交換実験で分かったデータ交換上の問題点と合わせて考えると、一定の交換レベルを設定し、その範囲内の構成要素を使用している図の交換を保証することが現実的である。

上記の問題点をふまえ、DXL規格改良案はコンフォーマンスレベル(以下CF)の考え方を導入する。

CFは木構造図が持つ概念の詳細度を、DXLが持つ概念の詳細度で表したものである。

各社木構造図の持つ概念、記述要素はまちまちで、全てが同じ詳細度、CFで揃うことはあり得ない。従ってどの様なCF構成が妥当か、標準化(WG11等)の場で検討して了解を得ていく必要がある。

SC7/WG11ではまず始めにCF-0、CF-1として

CF-0 ISO8631の構成要素と対応する詳細度

CF-1 CF-0を実用に向けて更に詳細化したもの(各木構造図の構成要素として了解が取れるもの)

を決め、DXLの新しい構文規則として表1の様にまとめた。

5 まとめ

木構造図 CASE ツール間でデータの交換を可能にするためにDXL規格案を作成し、データ交換実験の結果をふまえてその改良を行った。

DXL規格案は1992年よりISO/IEC JTC1/SC7/WG1(現在はWG11)の国際会議に提案し、規格化に向けて議論を行っている。本改良案を提案してISOでの規格化、及びJIS化を進めていく。規格化への課題としては、CF-0、CF-1に含まれなかった概念を扱うCF構成の検討や、交換ツール仕様の保証等が挙げられる。

謝辞 DXL規格案作成にあたった国内SC7/WG11の委員の方々、実験に参加して頂いた各企業のメンバーの方々に感謝致します。

参考文献

- [1] 平成3年度 ソフトウェア開発・システムの文書化標準化調査研究報告書、(財)日本規格協会 情報技術標準化研究センター、pp.11-49、平成4年3月
- [2] 志村武, 他: 木構造図用データ交換言語 DXL(2) - 相互接続実験 -, 情報処理学会, 第48回全国大会, 1994
- [3] 長野宏宜, 他: 木構造図用CASEツールのデータ交換言語 DXL, 情報処理学会誌 (掲載予定)

表 1: DXL 規格改良案の構文規則

[表記規則]	バックス記法に準ずる。
・	「または」を意味する。
[X]	0個または1個のXを意味し、{ X }は0個以上のXの繰り返しを意味する。
CF-0	の終端記号は7桁のラインを付け <u>aaa</u> 、非終端記号は <xxx> で表す。
CF-1	の終端記号は二重7桁のラインを付け <u>aaa</u> 、非終端記号は <<xxx>> で表す。
*****	字句要素 ***** (一部定義を省略) *****
<文字列>	::= [<1^ハイト文字> <2^ハイト文字>]
<書式制御文字>	::= ISO/IEC6429で定義
<コメント>	::= -- [<文字列> <書式制御文字>] <CR>
<識別子>	::= <文字列>
<仕様記述>	::= [[<文字列> <書式制御文字>]] [<<付加情報>>]
<<付加情報>>	::= % [[<文字列> <書式制御文字>]] %
<説明記述>	::= [[<文字列> <書式制御文字>]]
*****	DXL情報パッケージ *****
<モジュール名>	::= M_Packet <7^ローアイディアラフ> [<モジュール識別ラフ> <7^ロリスミアラフ>] End_M_Packet ;
<7^ローアイディアラフ>	::= Profile [<説明記述>] End_Profile ;
<モジュール識別ラフ>	::= Identification Identifier is <識別子> ; [<説明記述>] End_Identification ;
*****	アルゴリズムパラグラフ *****
<7^ロリスミアラフ>	::= Module_Algorithm <文の列> End_Module_Algorithm ;
<文の列>	::= { <文> }
<文>	::= { <<ラハ>> ; } <基本文> { <<ラハ>> ; } <7^ロク化文> { <<ラハ>> ; } <打ち切り文>
<<ラハ>>	::= <識別子>
*****	基本要素 ***** (ISO8631の5.1節に対応) *****
<基本文>	::= imperative [<<基本文タイプ>>] <7^ロク仕様記述> ;
<7^ロク仕様記述>	::= <仕様記述>
<<基本文タイプ>>	::= <<n u l l>> <<モジュール呼出>> <<g o t o>>
<<n u l l>>	::= null
<<モジュール呼出>>	::= call <モジュール識別子>
<<モジュール識別子>>	::= <識別子>
<<g o t o>>	::= goto <<ラハ>>
*****	構造をもつ要素の分類 ***** (ISO8631の5.2節に対応) *****
<7^ロク化文>	::= [[<7^ロク識別子>] [abstract <<7^ロク仕様記述>>]] <複合文>
<7^ロク識別子>	::= <識別子>
<<7^ロク仕様記述>>	::= <仕様記述>
<複合文>	::= <順次文> <並列文> <繰返し文> <選択文>
*****	順次要素 ***** (ISO8631の5.2節に対応) *****
<順次文>	::= begin <文の列> end ;
*****	並列要素 ***** (ISO8631の5.3節に対応) *****
<並列文>	::= parallel <文の列> end_parallel ;
*****	繰返し要素 ***** (ISO8631の5.4節に対応) *****
<繰返し文>	::= <前判定繰返し文> <後判定繰返し文> <継続繰返し文>
<前判定繰返し文>	::= condition [<<前判定繰返しタイプ>>] <条件仕様記述> loop <文の列> end_loop ;
<後判定繰返し文>	::= loop <文の列> end_loop condition [<<後判定繰返しタイプ>>] <条件仕様記述> ;
<継続繰返し文>	::= loop <文の列> end_loop ;
<条件仕様記述>	::= <仕様記述>
<<前判定繰返しタイプ>>	::= while until for
<<後判定繰返しタイプ>>	::= while until
*****	選択要素 ***** (ISO8631の5.5節に対応) *****
<選択文>	::= <単岐選択文> <多岐選択文> <多重岐選択文>
<単岐選択文>	::= if <条件仕様記述> then <文の列> end_if
<多岐選択文>	::= <列挙型選択文> <スイッチ型選択文>
<列挙型選択文>	::= exclusive_select if <条件仕様記述> then <文の列> [else if <条件仕様記述> then <文の列>] end_exclusive_select ;
<スイッチ型選択文>	::= exclusive_select case <評価仕様記述> [when <値仕様記述> ; <文の列>] end_exclusive_select ;
<多重岐選択文>	::= inclusive_select case <評価仕様記述> [when <値仕様記述> ; <文の列>] end_inclusive_select ;
<評価仕様記述>	::= <仕様記述>
<値仕様記述>	::= <仕様記述>
*****	打ち切り ***** (ISO8631の6節に対応) *****
<打ち切り文>	::= terminate <打ち切り対象> <仕様記述> ;
<打ち切り対象>	::= system module block <7^ロク識別子>