

# 交換ソフトウェア開発支援環境 (1)

## DDL開発支援システム

5S-2

川口 進 金子泰祥 丹野千秋 前島幸仁 \*  
 (株)日立製作所 戸塚工場 \* (株)日立製作所 中央研究所

### 1. はじめに

筆者等は、交換ソフトウェア開発における生産性向上のための重点施策として、データフローとデータ駆動論理をベースとした関式設計技法 (Data Driven Logic : DDLと以下称す) を交換ソフトウェアの論理的仕様記述に適用した。<sup>(1) (2)</sup>

本稿では、関式設計技法を採用した交換ソフトウェアの開発において、詳細設計からテスト工程までを一貫して支援する開発支援環境について述べる。

### 2. DDL技法

DDL技法は、詳細設計以降のタスクや分析フローのプログラム仕様をDDL図を用いてプログラムとして実現するものである。DDL図は、図1に示すデータフローダイアグラムで表現される。このDDL図において、基本的な演算を行う機能要素 (Function Element : FE) はノードで表し、その中に機能要素名を記述する。機能要素名は繰り返し使用されるので、それらを識別するための固有識別番号を付与している。また、各FE間で授受されるデータ (トークンと呼ぶ) の流れは、矢印付きアークで行き先を示す。

FEの演算処理は、そのFEの実行に必要な入力データが全て受け取られると当該FEが実行状態 (発火と呼ぶ) となり、その入力データに対して所定の演算が行われる。これをデータ駆動と言う。データ駆動の制御において、FEの演算結果が矢印付きアーク上にトークンとして流れ、次のFEへの入力データとなることで、計算が進行していく。従って、計算順序どおりに命令ノードを記述する必要はない。

### 3. DDL開発支援システムの機能構成

交換ソフトウェア開発のためのDDL開発支援システムの機能構成を図2に示す。本システムは、大きくホスト系とワークステーション系の2つに分かれている。ホスト系は、ホストコンピュータによるDDL図のコンパイル及び、テスト<sup>(3)</sup>を支援するシステムで、ワークステーション系は、ワークステーションによるDDL図の作成、表示、印刷を支援するシステムで、これらのホスト系と

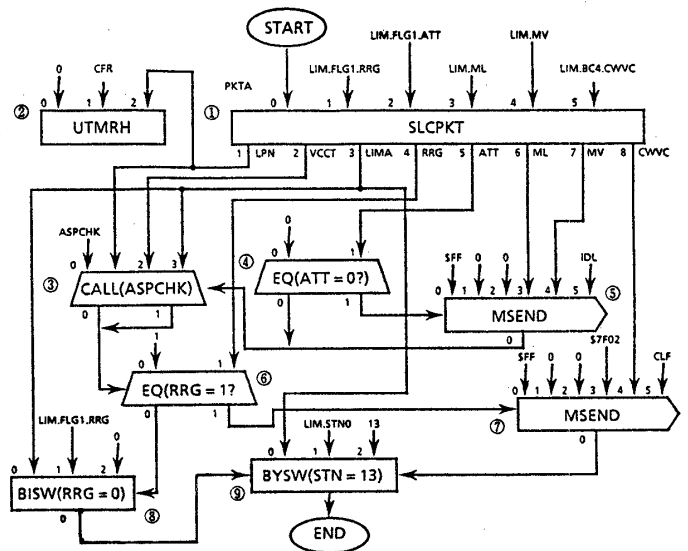
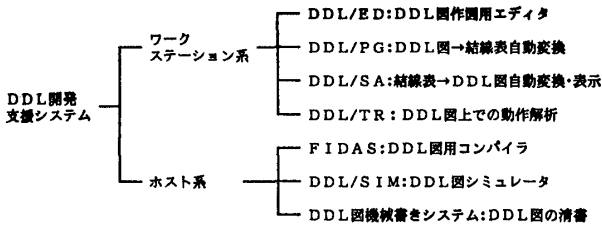


図1. DDL図例

ワークステーション系とを連動させることにより、DDL設計の下流工程、即ち、詳細設計からプログラミング、テストに至るまでの工程を一貫して支援する。

ホスト系は3つの機能から構成される。第1は、結線表から機械語に変換するコンパイラFIDAS、第2は、単体及び、機能レベルでのテストを支援するDDL/SIM (DDL/Simulator)、第3は、DDL図の結線表からDDL図ドキュメントをレーザービームプリンタに出力するDDL図機械書きシステムである。

一方、ワークステーション系は、次の4つの機能から構成される。第1は、マウスやアイコンによりノードやアークを用いて、DDL図を作成するエディタDDL/ED (DDL/Editor)、第2は、DDL図から結線表に自動変換するDDL/PG (DDL/Program Generator)、第3は、既存の結線表ソースコードからDDL図に変換し、ワークステーション上に表示するDDL/SA (DDL/Source Code Analysis)、第4は、DDL/SIMでのテスト結果に基づきDDL図上でビジュアルに動作解析するDDL/TR (DDL/Tracer) である。



ED: Editor PG: Program Generator SA: Source code Analysis TR: Tracer  
 FIDAS: Firing Design Automation System SIM: Simulator

図2. DDL開発支援システムの機能構成

DDL図に関する統計情報、例えば、修正履歴、設計履歴、DDL図の解析などに関する機能は、原則的には、ワークステーション系とし、結線表ソースコードやオブジェクトコードに関する管理支援機能は、ホスト系として機能分担している。

4. システム概要と作業の流れ

図3のシステムの概要に従って、DDL図の作成からテストまでの流れを示す。

(1) 詳細設計工程における作業

本工程での作業は、当社のワークステーション2050/32上で行われる。タスクや分析フローのプログラム仕様をDDL図で記述するために対話型DDL/EDを使用する。

DDL/EDでは、タスクや分析フローを記述するために専用画面を使用して、マルチウィンドウのもとで、アイコンやマウスによりメニュー登録された基本図形からFEノード、アークや接続子を選択し、画面上の任意の位置に配置した後、FE間の接続を行う。そして、DDL図は図形情報、例えば、ノードの形状、表示の座標位置、アーク数などを含む図形ファイルに格納される。次に、DDL/PGにより図形ファイルからDDL図と等価なテキスト形式の結線表に自動変換する。この結線表は、命令識別番号、入力ピン番号で記述し、結線の表現は入力ピンから出力ピンを参照する。そして、結線表はホスト側に転送され、結線表ファイルに格納される。

(2) プログラミング工程における作業

本工程での作業は、ホスト上で行われるが、設計者は作業指示を行うだけでコーディング作業は自動化される。即ち、FIDASはファイル単位に予め登録されているリテラル宣言や構造体定義を参照しながら、結線表を命令テンプレートに変換する。

(3) テスト工程における作業

本工程での作業は、DDL/SIMにより作成されたオブジェクトファイルを入力として、クロス環境下でDDL図をデータ駆動に基づいて実行する。この過程で取

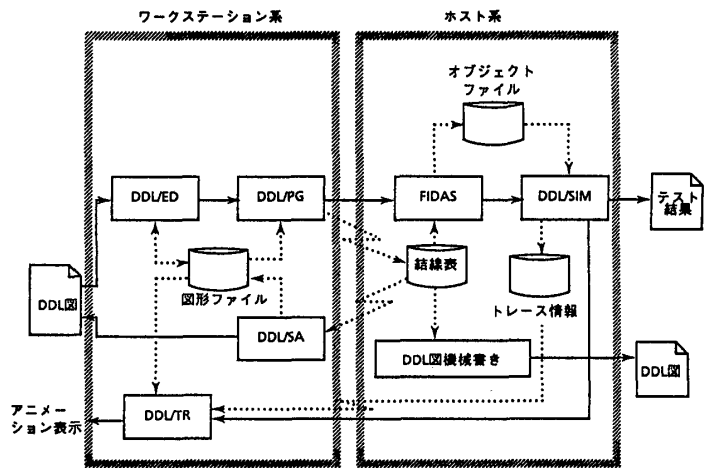


図3. システム概要

集される命令の実行順序や命令の入出力データのトレース情報は、トレース情報ファイルとラインプリンタに出力される。次に、トレース情報ファイルに格納された結果を2050/32上のDDL/TRに転送し、図形ファイルの図形情報と組み合わせ、2050/32上にDDL図を表示する。ここで、設計者は、表示されたDDL図上でアニメーション表示しながら、DDL動作の正常性をチェックする。

また、新機能追加によるプログラム修正がある場合には、修正対象となるプログラムが既存機能に対して、影響を与えるかどうかを検証する必要がある。この場合には、予め登録されている正しいトレース結果と新機能追加後のトレース結果をDDL/TRに転送し、DDL/TRが両トレース結果を比較することにより、自動的にディグレード検証する。

5. おわりに

交換ソフトウェア開発支援環境としてDDL開発支援システムを開発し、当社電子交換機CX5000シリーズの呼処理プログラムの詳細設計以降の工程に適用し、品質及び、生産性向上の点で効果を上げている。

[参考文献]

(1) SHIRASU H.,1987,:"Innovative Approach to Switching Software Design Using Data Flow Concept", ISS-87,2,B-4-3  
 (2) 前島、外: データ駆動論理(DDL)プログラミングにおける支援環境、電子通信学会 交換研究会 SE87-144 pp. 79-84, 1987  
 (3) 川口、外: 呼処理プログラムシミュレータ 電子通信学会 交換研究会 SE88-86 pp. 43-48, 1988