

New-SWB プログラム設計支援ツール/MCDtools

3M-7

藤本繁喜, 内金崎誠一, 仙田和彦, 建部周二
(株)東芝 府中工場

1. はじめに

東芝府中工場で推進しているソフトウェア生産支援システムNew-SWB^[1]でのプログラム設計の考え方と支援ツールについて説明する。

2. MCDによる設計手法

ソフトウェア開発について様々な手法が示されてきた。MCD (Module Connection Diagram) は、それらを取入れた開発手法である。考え方は以下の通りである。

- (1) 構造化設計による、モジュール化、トップダウン開発と、経験や標準ライブラリに基づくボトムアップ開発の融合。
- (2) 大規模リアルタイムシステムの開発への対応。
- (3) 設計に必要な概念と手順と評価方法の標準となる。

2. 1 概念の整理

MCDは、その名が示すようにモジュールの組合わせでシステムを表現するダイアグラムであり、MCD手法ではモジュールの組合わせでシステムが構成されることを基本とする。モジュールとはソフトウェアシステムを構成する論理的な構成単位である。計算機システムの構成モジュールは、下記のように分解される(図1)。

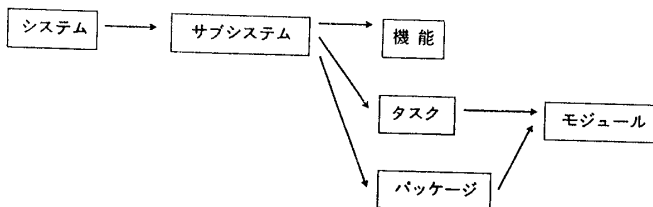


図1 ソフトウェアシステムの論理階層

大規模リアルタイムシステムにおいて、一つのシステムは、独立した複数のサブシステムに分解される。各サブシステムごとに、まず機能ダイアグラム(FCD)を用いて機能とデータの関係在设计する^[2]。これは、要求仕様を受けて先ず論理的なソフトウェア設計モデルを作る過程である。

これにたいしMCDを用いる工程は、実際の計算機のOS上で動くモデルの設計を行う工程である。

MCDで扱うモジュールとは、下記の3種類である。

- (1) タスク サブシステムを構成するモジュールで平行して実行可能な処理単位である。
- (2) パッケージ サブシステムを構成するモジュールで共通ファイルやテーブルへのアクセスを行うデータアクセスパッケージと、共通の機能を集めたライブラリパッケージからなる。
- (3) プログラム・モジュール タスクを構成するモジュールで、プログラムのサブルーチンやプロシジャに相当するもの。単にモジュールとは以後、これを指す。

2. 2 設計手順

New-SWBでは、MCDによるプログラム設計の前に、FCDによりサブシステムの機能分解とデータフロー及びコントロールフローを、明らかにする。MCDでの設計はこれ等の情報をもとに漏れなくそれらの実現形式を決めることにある。

(1) サブシステムのタスク、パッケージへの分解

- 入 力: FCDの機能分解
出 力: 機能とタスクの相関関係(どのタスクがどの機能を実現するか)の定義
タスク外部仕様、パッケージ外部仕様
タスクとパッケージ間のコントロール・データフロー(H-MCD 図 次頁参照)

(2) タスクのモジュールへの分解

- 入 力: (1) の出力
出 力: タスク内モジュール階層図(V-MCD 図)
モジュール外部仕様

(3) パッケージのモジュールへの分解

- 入 力: (1) の出力
出 力: パッケージ内モジュール階層図(P-MCD 図)
モジュール外部仕様

(4) モジュールの詳細化

- 入 力: (2) または(3) の出力
出 力: モジュール内部構造図(フローチャート、TFP、PAD)

(5) ソースコード作成

- 入 力: (4) の出力
出 力: ソースコード

New-SWB Program Design Support Tools/MCDtools

Shigeki Fujimoto, Seiichi Uchikanezaki, Kazuhiko Senda, Shuji Tatebe

TOSHIBA CORPORATION

2.3 設計の評価

この様な設計手順で作られる設計書について、以下の確認が必要である。

- (1) 表現が統一されていること
- (2) 上流と下流工程間で抜け、漏れの無いこと
- (3) 各設計書の内容が相互に矛盾しないこと
- (4) 第三者が用意に検査できること

3. MCDtools

MCDtoolsは、MCD手法の概念、表記法、手順、評価を支援するツールで、当社のAS3000上のマルチウィンドウ環境にインストールされている(図2)。

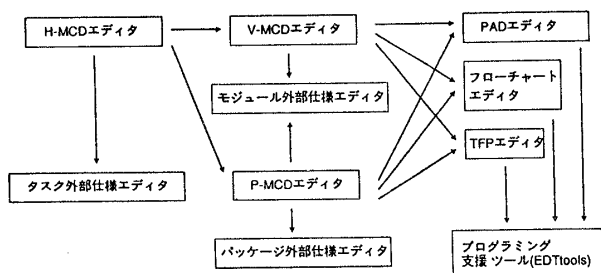


図2. ツールの構成と相互関係

(1) 編集ツール

設計手順に示した各種設計図を作成編集するツールである。

図3は、H-MCDとそのための専用エディタを示している。たとえば、白い箱はタスクを、箱の隅に黒い三角形のマーク付きのものは、パッケージを示す。ユーザインタフェースは、右側に設計モジュールを上側に編集アイコンを配した対話形式の作画ツールである。

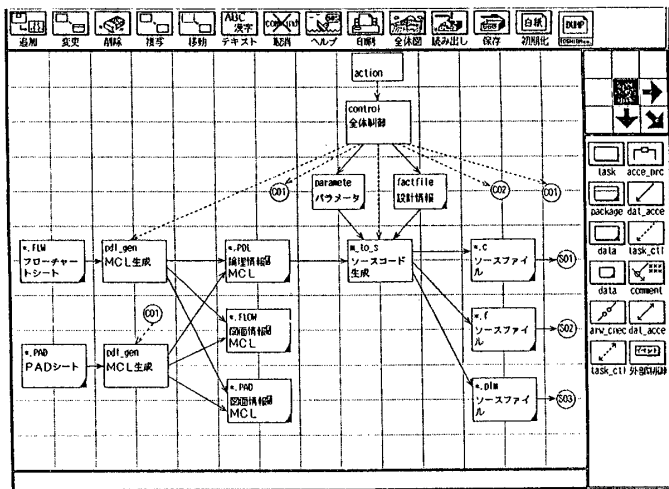


図3. H-MCDとそのエディタ

(2) 変換ツール

チャート間の変換(フローチャート、PAD、TFP)やソースコード→チャート、チャート→ソースコードの自動変換を行なう。

(3) 設計情報の格納

設計データディクショナリに編集ツールの作成した設計情報の格納ができる。これを利用して、各設計ツール間の情報の一貫性を保証する。

(4) 設計手順の支援

画面遷移機能により、ある設計モジュールを画面上で指定してそのモジュールの詳細設計をみるといった詳細情報の参照が容易にできる。これは、Hyper Text的な操作の実現である。

図4に画面遷移の例を示す。このようにH-MCDからV-MCDへ更に個々の設計モジュールの仕様へと次々と指定設計モジュールの詳細情報を呼出すことができる。

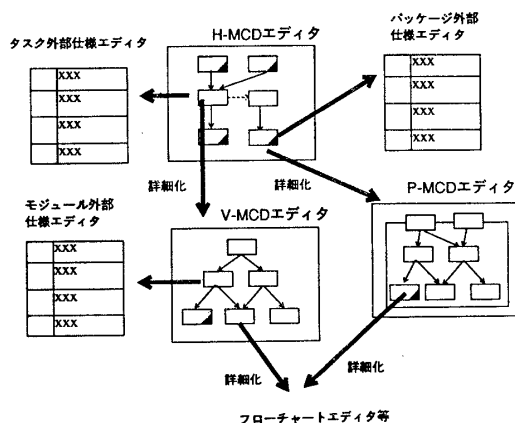


図4. 画面遷移の例

4. 効果と今後の検討

これらツールの適用により手書き設計に比べて40%の生産性向上が達成でき、かつ本ツールが目指す構造化設計支援の作業環境が構築できた。表現方法や手順については自由度が高いツールとなっており、今後適用事例を増やして、標準部品の蓄積と、ガイド機能の検討をはかっていく予定である。

参考文献

1. New-SWB 大規模リアルタイム・ソフトウェア開発環境、小野他、情報処理学会 第37回(昭和63年後期)全国大会
2. EWSによるシステム設計支援ツール(1) --機能設計支援一、小林他、情報処理学会 第34回(昭和62年前期)全国大会 1S-3.