

New-SWB プログラミング支援ツール/EDTtools

3M-8

建部周二、溝淵順子、荻原啓孝、横山信行
(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]のプログラミング支援の考え方とツールについて説明する。

2. New-SWBでのプログラミング作業

プログラミングの環境について、高級言語の導入、構造化プログラミング、最近の新しいパラダイムに基づく言語の展開などの言語の側面と、コンパイラ、デバッグ、エディタなどのツールの二つの側面がある。

さらに開発環境全体では、まずプログラミングをどう位置付けるかという問題がある。

(1) プログラミングの位置付け

プログラミングということばは、単にコーディングだけでなく、プログラムの構造を決める設計作業も含める方が多い。個人的、もしくは一過性の作業であれば、全てプログラムコードで行われることもある。

New-SWBでは、プログラムの構造は、MCDで設計し、詳細のプログラム構造とコードをプログラム言語で記述する。この両者の情報は、ツールで統合する。これは、プログラムの細かいロジックや見た目のコーディングだけを見るのではなく、全体の構造を設計モジュールの組合わせとしてとらえようとするものである。

(2) 言語

当工場内で広く用いられかつ互換性と開発従事者数の高いFortran, C, PL/Mの3種類の言語を標準としている。これらの言語をそのまま用いて、構造化コーディングの規則を定めている。

Adaや知識処理言語等の新言語の導入や、プログラミング言語の設計言語への上位拡張はまだ一般的ではないが検討を開始している。

コーディング作業は、MCD手法で決まったプログラムの構造を更に詳細に展開する形で行われる。

(3) ツール

既存のコンパイラ、デバッグやunixの各種ユーティリティを構文エディタを中心に利用できるようにして、多種多様のターゲットの要求に答えられるようにしている。

それらがバラバラにメニュー化されてるのではなく、MCD手法を支援するMCDtools^[2]の作り出す設計情報をもとに各ツールがまとまって使えることと、逆に例えばチャートエディタのボックスでしかコーディングできないといった制約も、できるかぎりはずすことを目標とした。

3. ツールの構成

EDTtools^[3]は、New-SWBのプログラミングを支援するツールであり、図に示すサブシステムより構成される。当社のEWSであるAS3000上に、他のツールと統合化されて搭載される(図1)。

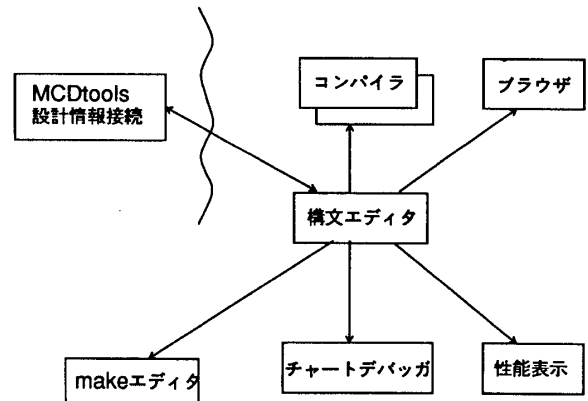


図1. ツールの構成

(1) 構文エディタとブラウザ

Fortran, C, PL/Mの構造化コーディングを支援するツールである。他のプログラミング支援ツールを起動する。

(2) コンパイラ

多種多様な実機がある場合、コンパイラは、クロスコンパイラをホスト機上に搭載するか、実機そのものを利用するかのどちらが多い。EDTtoolsでは、共用のコンパイラマシンをEWSにLAN接続する方式をとった。

現在、TCP/IPプロトコルを用いてこれを実現している。

(3) ロードモジュール作成

MCDtoolsのV-MCD図(タスク内部仕様)により、各タスクのモジュール構造を設計する。また、P-MCD図(パッケージ内部仕様)により、各ライブラリの構成方法が設計されている。

EDTtoolsでは、この情報をもとにどのソースコードを組合わせてコンパイルをかけ、オブジェクトを組立てるべきか決定する。unixのmakeファイルをMCDから自動生成してこれを実現している。

(4) デバッグの支援範囲

V-MCDで設計するひとまとまりの単位、即ちタスク単体までの試験をEWS上で出来るようにビジュアルなデバッグとパスカバレッジ測定を行なう。

組合わせ試験と総合試験のフェーズは、様々な条件設定が必要であり、実機での試験となるため、現在のEDTtoolsの範囲には含まない。

4. 作業の流れ

まず、MCDで設計が終わるとEDTtoolsに対して下記の形で情報が準備される(図2)。

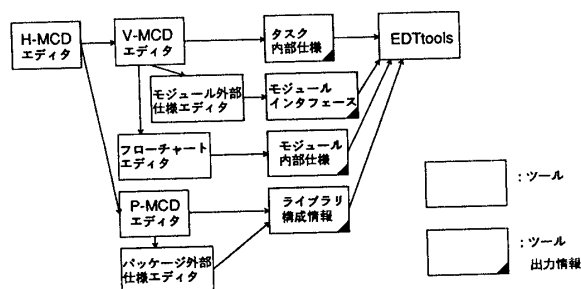


図2. EDTtoolsに渡される設計情報

(1) V-MCDからのEDTtoolsの起動

ダイアグラム上の指定したモジュールに対しコーディング要求を行う。構文エディタが起動される。

(2) コーディング

構文エディタは、指定モジュールのスケルトンが入った形で起動されている。エディタのコマンドを用いて、これを補う形でコーディングを詳細化する。

(3) コンパイル

コンパイルボタンをクリックすると、現在編集中のソースコードに対して、適切な言語及びターゲットのコンパイラが起動される。結果はコンパイルウィンドウに出力する。

(4) ロードモジュール作成

makeエディタを起動し、適切なmakeが生成していることを確認後、リンクを実行する。

(5) デバッグと評価

V-MCD図と各モジュールのソースコードを表示しながらデバッグを行う。結果のパスカバレッジをV-MCD図上、もしくは一覧表でレポートする。

図3は、以上の作業の流れを示している。

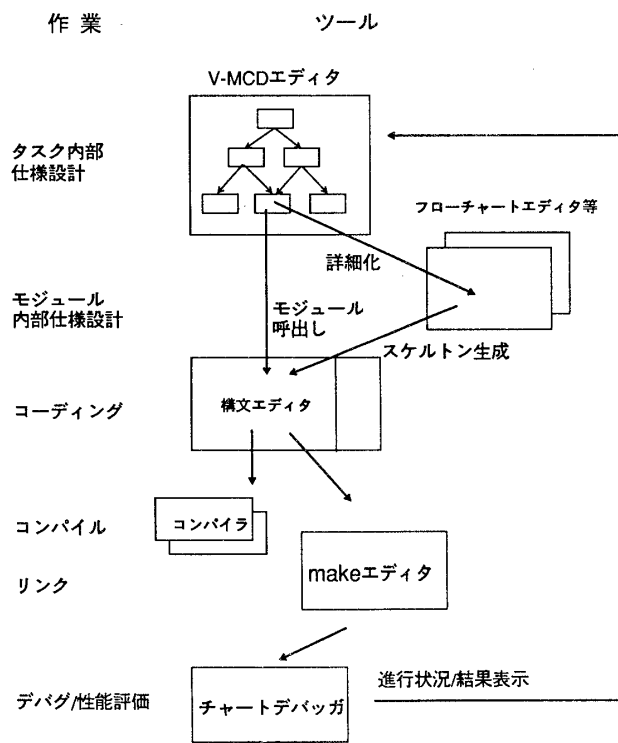


図3. 作業の流れ

5. 評価

MCDtoolsと組合わせて従来に比べて30%の生産性向上ができ、かつ構造化設計に基づくプログラミング・単体試験と、各種プログラミングツールのメニュー化ができた。

今後、適用事例の蓄積と、作業ガイドの検討、新言語の検討、さらに組合わせ試験以降の支援が課題である。

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. New-SWB プログラム設計支援ツール/MCDtools 藤本他、情報処理学会 第37回(昭和63年後期)全国大会
3. 構文エディタと部品検索ブラウザによるプログラミング支援ツール、建部他、情報処理学会 第36回(昭和63年後期)全国大会 2L-6