

「構造」編集・処理システム *nnHEADWAY* の開発

6D-2

水谷 高康 Peter Thoeny

日本電装クリエイト株式会社

1 はじめに

質の良いソフトウェアを開発するためには、構造化という概念が必須のものとなっている。現在用いられている各種の構造化技法は、ソフトウェアのライフサイクルに含まれる各工程あるいは全体を対象にしている^[1]。この構造化の対象を拡大すると、管理業務の構造化など、さまざまな方面に適用することができる。我々は、この「構造」に注目し、「構造」の編集・処理を行うシステム *nnHEADWAY* を開発している。

2 構造

本システムでは、構造を持つものすべてをシステムの対象とする。ここで扱う「構造」は、以下のように定義される。

構造： ある要素を、それが構成される1つ以上の細かな要素で表現できるもの。

これによって、プログラムの制御構造・データ構造などソフトウェアの設計・開発はもとより、プロジェクト管理など多彩な分野での設計・管理・実行を本システムで扱うことができる。

3 構造の表現

本システムでは、構造をSCD(Structured Chart Design)またはHCPチャート^[2]を使用して表現する。図1に、構造の表現例を示す。

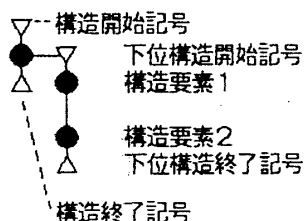


図1 構造の表現

SCDでは、以下のようにして構造を表現する。

- (1) 構造は、構造開始記号・構造要素列・構造終了記号によって表現する。
- (2) 構造要素列は、構造要素を0個以上ある順序にしたがって並べたものである。
- (3) 構造要素は、その適用方法によって2つの種類に分類される。
 - (A) 通常要素

一度だけその要素が適用される。通常要素には以下のものがある。

- ・正規要素
 - 一度だけその要素が適用される。
- ・参照要素
 - 別に定義されている構造をその場所で参照し適用する。
- ・マクロ要素
 - 別に定義されている構造を要素の下位構造とする。
- ・チェック要素
 - 何らかのチェックを表す要素である。
- ・疑似要素
 - 適用が行われない要素である。

(B) 適用制御要素

必ず下位構造が存在し、その下位構造の適用方法を制御する。

- ・選択要素
 - 選択構造を下位構造に従え、そのうちの幾つか(多くの場合1つ)を条件によって選択し適用する。
- ・繰り返し要素
 - 下位構造を0回以上繰り返し適用する。
- ・並列要素
 - 選択構造を下位構造に従え、その各構造を並列に適用する。

- (4) 各構造要素がさらに構造を持つ場合は、その要素に下位構造に従えることができる。
- (5) 選択構造は、複数の構造要素列を選択枝として持つ構造である。

4 *nnHEADWAY*

4.1 システム概要

本システムは、MS-Windows, OS2/PM, OSF/Motifなどのウィンドウシステム上で動作する。*nnHEADWAY*の画面は、2つのウィンドウ・3領域で構成される(図2)。

(1) チャートウィンドウ

- ・構造化チャート領域
 - マウス・キーボードによって構造化チャートを編集する。
- ・1次テキスト領域
 - 構造の各要素の説明文や、構造を処理する場合の処理内容を記述する。

(2) 2次テキストウィンドウ

- ・2次テキスト領域
 - 構造を処理した結果発生するデータや、構造の各要素に対応するデータを表示/記述する。

The development of *nnHEADWAY*, a system to edit and process structures.

Takayasu Mizutani, Peter Thoeny

NIPPONDENSO CREATE Co., Ltd.

2-15-20, Nishiki, Naka-ku, Nagoya 460, Japan

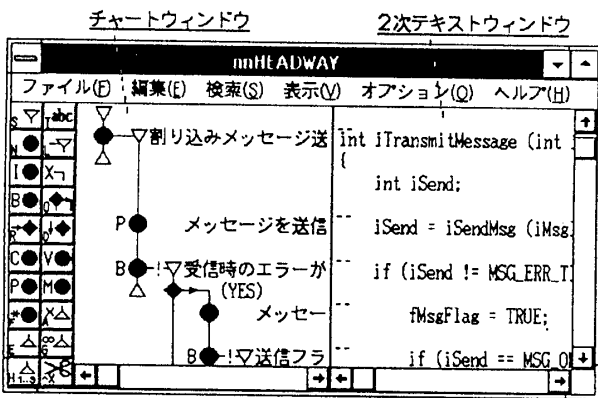


図2 nnHEADWAY表示画面

4.2 機能

本システムの主な機能を以下にあげる。

(1) 構造化チャート編集機能

マウス・キーボードによって構造化チャートを編集する。ウィンドウシステムの GUI を用いて操作性の良いユーザインタフェースを提供する。

(2) 構造処理機能

1次テキストに記述された処理内容を解釈し、構造を適用する。処理内容は、ユーザが自由に記述することができる。

(3) テキスト編集機能

1次テキスト・2次テキストを対象にテキストエディタとしての機能を備えている。

(4) ファイル出力機能

本システムでは、3つの領域に記述された構造やテキストのデータを統合して1つのファイルとして管理する。また、2次テキストデータを独立したファイルに出力する機能を持っている。

4.3 構成

本システムの構成を図3に示す。

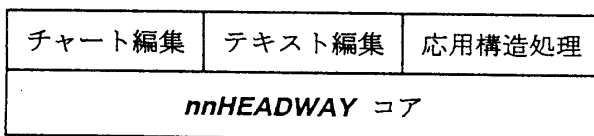


図3 システム構成

nnHEADWAY コアは、ウィンドウマネージャからのメッセージハンドリングや、基本構造処理、画面の描画などの処理を行う。

本システムは、構造処理機能を使用目的に合わせて変更することにより、さまざまな用途に使用することができる。

5 nnHEADWAY/P

ここでは、本システムをプログラム開発に適用したアプリケーション nnHEADWAY/P について述べる。

nnHEADWAY/P は、ソフトウェアのライフサイクルのうち、設計フェーズとコーディングフェーズをサポートする。

5.1 プログラム開発対応

本システムの画面上の3つの領域を以下のように使用する。

(1) 構造化チャート領域
構造化チャート領域には、プログラムの制御構造を記述する。

(2) 1次テキスト領域
1次テキスト領域には、構造化チャートの説明文を記述する。

(3) 2次テキスト領域
2次テキスト領域は、ソースコードを記述する。

つまり、チャートウィンドウの内容が設計書に、2次テキストウィンドウの内容がソースコードになる。このように対応させることによって、設計書とソースコードを一元管理することができる。

5.2 特徴

nnHEADWAY/P 独自の機能として以下の特徴を持っている。

(1) 開発補助機能

2次テキストに記述されているソースコードを単独ファイルとして自動抽出し、コンパイラ・リンク・メイクなどの開発ツールを呼び出す。

(2) チャートの自動生成

C言語で記述されたソースファイルを解析し、構造化チャートを生成する。このとき、同時にソースファイルを2次テキストとして取り込む。

6 おわりに

ソフトウェアの設計・開発だけでなく、幅広い分野を対象を拡大した「構造」編集・処理システム nnHEADWAY について概説した。また、このシステムをソフトウェアの開発に適用した nnHEADWAY/P について紹介した。

今後は、本システムを基本にして各分野の設計・管理を行うシステムを開発する。

参考文献

[1] Carma McClure : C A S E, 日経BP社 (1990).
[2] 花田收悦 : プログラム設計技法, 企画センター (1983).