

# ソフトウェア開発支援システム

山本 貴博 大岩 元  
(豊橋技術科学大学)

## はじめに

ソフトウェアの開発・設計において、初期段階での論理的な構造の決定は、非常に重要なものである。このための設計図法として、我々は、HCPチャート【1】を採用している。本システムは、計算機上でHCPチャートの記述を可能にし、さらにHCPチャートとソースプログラムを結合させることで、ソフトウェアの開発・保守を支援するものである。

このために、HCPチャート用のチャートエディタを作成する。処理記述の日本語入力方式として、TUTコード【2】によるタッチタイプ方式【3】を採用することでさらに、チャートの記述を容易にする。

## 目的指向設計

目的指向設計とは、問題の目的を明確にさせながら論理的な構造を整理して詳細化を進めて行く設計方法である。

問題（要求）から実現への基本形態がどうあるかと考えると、

1. 「何をするのか」 (what)
2. 「そのためにはどうやるのか」 (how)

この二つの繰り返しで問題を詳細化していく。ここで重要なのは、現実手段に目を向ける前に「その目的は何か」「意味は何か」...を常に明確にしながら設計し、整理していくことである。これは、筋道をはっきりさせ論理的な誤りの侵入を防ぐのに役立つ。このように、処理目的と実現手段の対応付けを階層的に構築して、目的の明確化・詳細化を行ないながら設計を進めていく設計方法が目的指向設計である。

## HCPチャート

目的指向設計の設計図法として、HCPチャートを用いる。

HCPチャート(Hierarchical and Compact Description Chart)は電電公社横須賀電気通信研究所で開発された設計図法であり、処理目的(what)とその実現手段(how)とを対にして階層的に処理の詳細を記述するものである。チャート上には、処理の記述(論理的な処理の階層、制御構造)、データの記述(論理的構造、インタフェース)が書かれる。

HCPチャートの記述は、

「...をする」  
そのためには「...をする」

という具合に、自然言語によって記述していくので、設計に集中しやすく、更に他人が読んでも分かりやすい(図1)。

また、HCPチャートは、設計記述が直接プログラミング構造に反映されるので、ソースプログラムの保守用ドキュメント(論理構造のドキュメント)として使用される。

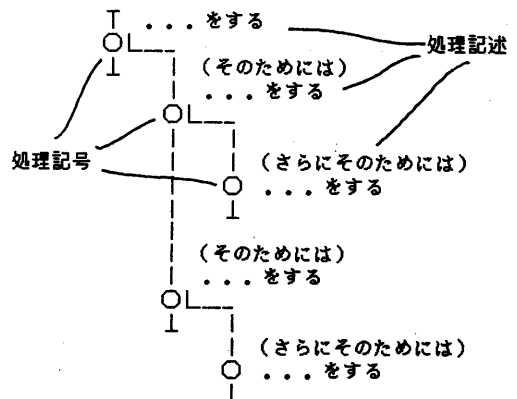


図1. HCPチャートの書き方

HCPチャートは原則的に手書きである。しかし、処理記述が込み入ってくると修正が大変であり、何度も書き換えが必要となってくる場合が起きてくる。このため、チャートがメモで終わってしまう可能性が大きい。こうなると、チャートとプログラムの内容が合わなくなり、チャートのドキュメントとしての意味がなくなる。

このため、オンラインでのHCPチャートの記述が必要となる。さらに、チャートとソースプログラムを同一のシステムで一体化して記述することを考える。

#### システムの特徴

本システムの特徴として以下のことがあげられる。

##### 1. 操作が簡単である。

チャートをオンラインで記述する場合で重要なことは、設計者（システム使用者）の思考を妨げないことである。つまり、できるだけ簡単な操作で記述できなければならない。このため、チャートエディタであってもキャラクタを扱うスクリーンエディタの手軽さが必要となる。本システムは、このことを重視して設計している。

##### 2. 処理説明の記述をタッチタイプで入力できる。

処理記述が能率的に入力できることは、本システムの目的からも重要である。そこで、処理記述の日本語入力方式として、高速入力が可能で疲労が少ないTUTコードによるタッチタイプ方式を採用する。

以下は、TUTコードの特徴である。

1. 2ないし3ストロークで、ひらがな・漢字合わせて約2500文字を直接入力できる。
2. コードの割り当ての最適化を行なってお

り、高速入力が可能。

3. ひらがなについては、構造的にコードが割り当てられており、覚えやすい（3時間でブラインドタッチ入力が可能）。従って、カナ漢字変換を用いて直ちに高速入力ができる。

3. チャートとプログラムを一体化できる。  
システム内でソースプログラムを入力できる。HCPチャートの最下位レベルの記述に実際のソースプログラムを埋め込む。

HCPチャート上の処理記述とソースプログラムを適当に出力することで、論理構造（HCPチャート）とソースプログラムの対応を明確化させることができる。

##### 4. ターミナルを選ばない。

オンラインでチャートを記述させるために、日本語グラフィックスターミナルを対象としてシステムを作るが、一般性を持たせるために、キャラクタターミナルでも記述できるようにする。本システムでは、ANSIIに準拠した制御コードを持つターミナルを対象とした。このために、HCPチャートの特徴である円を基本とした記号を表現することは出来ない。しかし、この記号は手で記述する事を前提として設定したものであるから、オンラインでチャートを記述させる場合には、他の表記でも問題とはならないであろう。そこで、通常の処理は英字の"SE"で表し、繰り返し処理は"RP"で表すというように、英字で代用させる（図2）。

#### システムの構成

本システムは、UNIX上で開発・動作している。開発したシステムの記述言語はCである。

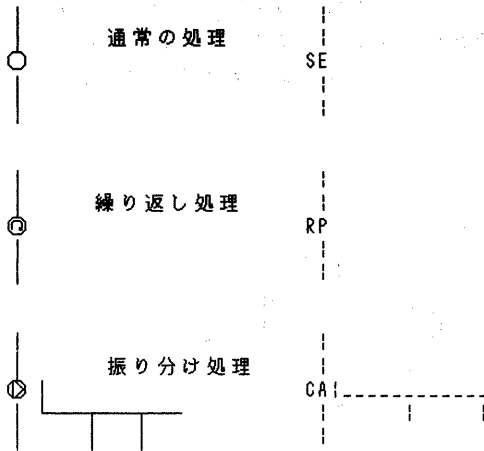


図2. 処理記号の出力例  
 (左: グラフィックスタターミナル  
 右: キャラクターミナル)

データ構造

HCPチャートは階層構造を重視した設計図法であるから、この階層構造を十分表現出来るような内部データ構造としなければならない。そこで、次の基本データ構造(セル)を考えた(図3)。各処理記号記号とそれに対応した処理記述は、このセルにより表される。

接続方向・階層方向のポインタは、各々の記号をリンクするためのもので、各セルは、双方向のリンクでつながっている(図4)。記号は、処理記号のコードが入る。処理記述へのポインタは、処理記述の文字列を指している。プログラムへのポインタは、関連するソースプログラムを指している。X・Y座標は記号の画面表示のための内部座標が入る。

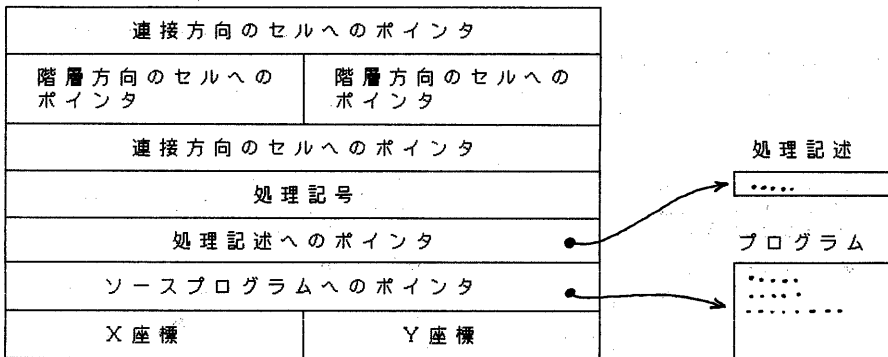
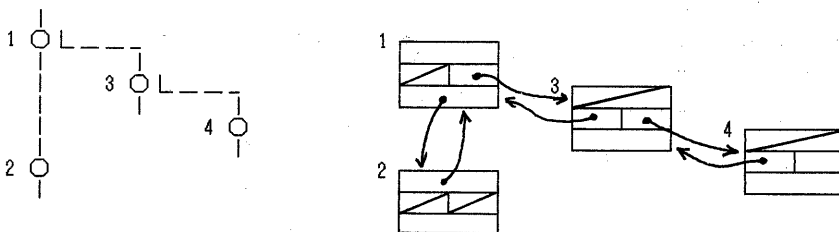


図3. 基本データ構造(セル)



(セルの階層・接続方向のポインタだけ示す。番号は、対応する処理記号とセルを指す。)

図4. セルのリンク例

**内部コード**

処理記述は、日本語データに対してはJISの2バイトコードを用いる。ただし、ASCIIコードしかサポートされていないターミナルに対しては、ASCIIコードをそのまま内部コードとしている。プログラムも同じく、ASCIIコードを内部コードとして用いる。

**編集コマンド**

各編集コマンドは、接続方向と階層方向の二つの処理方向を持つ(図5)。この方向は、モードを切り替えることで変えることができる。

カーソルは、モードに従って、接続方向と階層方向に移動することが出来る。

挿入処理についても同様に方向に依存させた形を持たせる(図6)。

削除については、削除しようとする対象が、よりレベルの低い処理を持つなら、これ以下の

レベルのものを全て削除する(図7)。削除した情報はバッファに蓄えられる。これは、putコマンドにより、任意の位置に取り出せる。

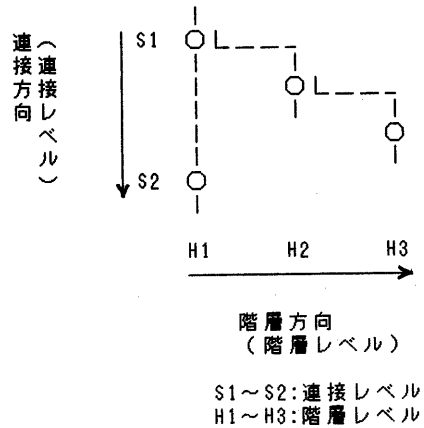


図5. 接続・階層方向(レベル)

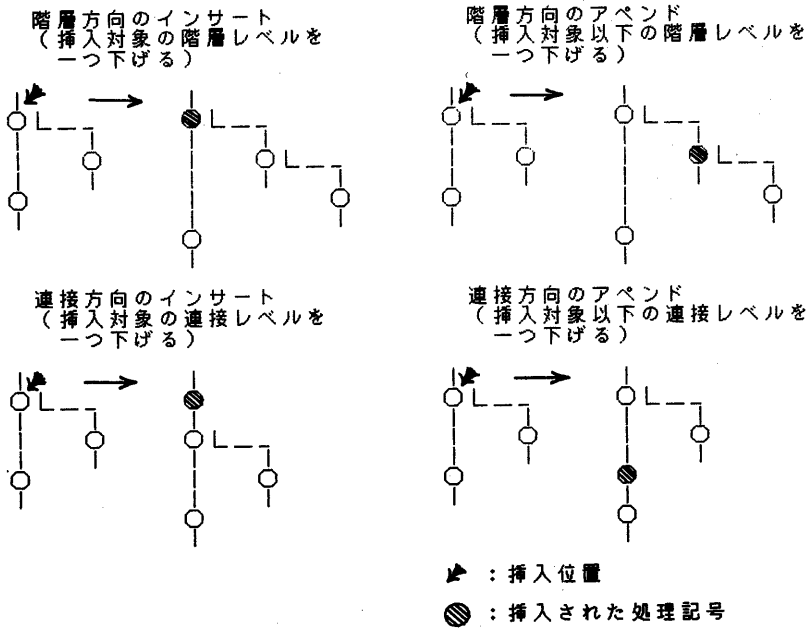


図6. 挿入処理

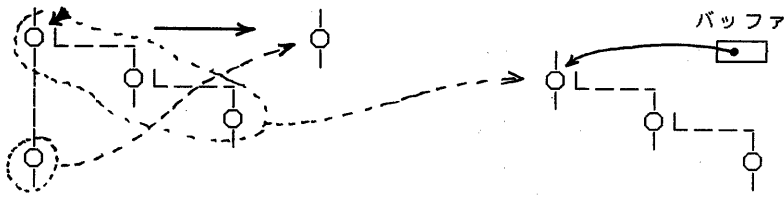


図7. 削除処理

#### 編集コマンドの例

^ V	接続方向へのモード切り替え
^ H	階層方向へのモード切り替え
a	アペンド
i	インサート
d	削除
p	バッファからのコピー
j, k, h, l	カーソル移動
cc	処理記述の変更
cs	処理記号の変更
: r	ファイルの読み込み
: w	ファイルの書き込み
: e	プログラムの入力

以上の基本コマンドは、UNIXのスクリーンエディタviを基にしている。

#### フォーマットと画面表示

コマンド実行の際、全体を作図し直すのでは実用にならない。そこで、編集作業に応じて影響を受けるセルは自動的に書き直すようにする。

このため、まず各階層レベルでの最大処理記述長を求める。これによりX方向が決定される。つぎに、それぞれのセルに対して、それに関連するセルの個数を求める。これによりY方向が決定される。最後に、各セルに座標を書き込む。

#### ファイル入出力

HCPチャートファイルには以下の情報が出力される。

- 各セルのリンク情報
- 処理記号
- 処理記述
- ソースプログラム

このファイルは、読み込み時に次の処理がなされる。

- 必要なセルを発生させる。
- 各セルをリンクする。
- フォーマットして、画面出力する。

#### HCPチャートとソースプログラムの結合

チャートエディタの中でソースプログラムを、チャートに添って入力できるようにする。

結合されたプログラムは、次の二種類の形で出力させる。

1. プログラムと処理記述を注釈として合わせたもの（リスタンピング用）。
  2. プログラムだけのもの（コンパイル用）。
2. は、JISコードを受け付けられないコンパイラのために必要となる。

### おわりに

HCPチャートの問題点として、データ構造の表現が十分でない。さらに、内部データをチャート上に記述すると、チャート全体の明確さが失われやすい。このため、本システムでは、データ構造及び内部データの表現は、別のドキュメントで扱う。

### 参考文献：

- 【1】 花田収悦「プログラム設計図法」企画センター（1983）
- 【2】 高嶋孝明「タッチタイプ入力の一方式とその練習法」情報処理学会日本文入力方式研究会資料4-3（1982）
- 【3】 大岩 元（監修）「タッチタイプの本」エー・アイ・ソフト（1984）