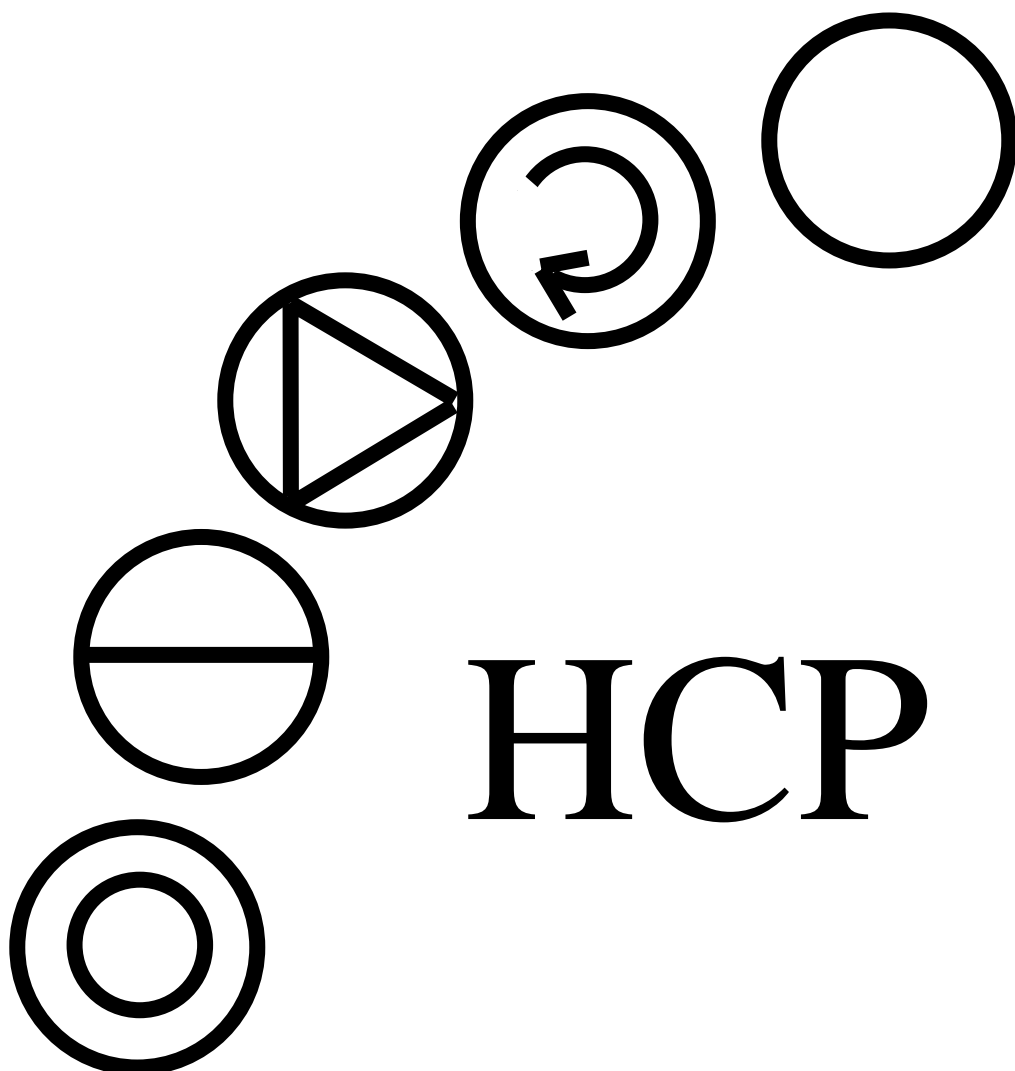


HCP チャートによる ソフトウェア開発

株式会社デンソークリエイト

山本雅基 執筆



HCP

株式会社デンソークリエイト

はじめに

腕の立つ料理人が高度な技を持ち良い道具を使っているように、プロのソフトウェアエンジニアも高度な技術力を持ち良い道具を使っています。プロのエンジニアは自分たちの高度な技術力を使って、良いソフトウェアを次々に開発します。時として、困難な要求仕様や厳しい納期にさらされますが、良い道具（プログラム開発ツール）が彼らを助けてくれます。

古く錆びた包丁で魚を三枚におろすのが大変なように、時代遅れのプログラム開発ツールで素早く良いソフトウェアを開発することは難しいことです。特にソフトウェアの分野は技術の進歩が非常に早く、開発ツールがすぐ時代遅れになることがあります。反面、技術の進歩が早いため、少し前まで高かったり使いにくかったりした開発ツールが簡単に手に入るようにもなります。

この本では、ソフトウェア開発のためのとても便利な道具である HCP チャートを用いたプログラム開発方法を紹介します。

HCP チャートは、1980 年代の始めに日本電信電話公社（現 NTT）横須賀電気通信研究所で開発され、国際規格として ISO/TC97/SC7（ソフトウェアの開発とシステムの文書化）でも議論された構造化プログラム設計のための図式表現法です。発表当時からその有効性が多くの人に認められ、プロのソフトウェアエンジニアの間で好んで使われてきました。そして、この 10 年余りの間に HCP チャートを作図する便利なチャートエディタが販売されるようになり、ますます使いやすくなってきています。今や、HCP チャートは構造化プログラミングを助け、ソフトウェア開発のレベルを押し上げる、本当に重要なチャートとなっています。

構造化プログラミングという考え方は 25 年も前に提唱され、その良さは十分に立証され、今では大変広い範囲で活用されています。現在でも、新しいソフトウェア開発の方法として、この構造化プログラミングを基本として様々な提案がされています。この分野に関する研究は、ソフトウェア開発論として各学会などで盛んに行われています。それでもまだ、構造化プログラミングに踏み込めずにいるエンジニアも多いようです。そのような方こそ、本書を読んで HCP チャートに慣れ親しんで、自らの技術に新風を吹き込んで欲しいと思います。HCP チャートは構造化プログラミングの有効な道具なのです。

ソフトウェア開発の現場では、研究中のソフトウェア開発論をリスクを犯して使って納期の厳しい仕事をするわけにはいきません。本書では、十分に実績のある HCP チャートを用いたソフトウェア開発方法について述べます。実は、HCP チャートを用いることで、自然に品質の良いソフトウェア開発を行うことができることが、私たちの経験上分かっています。ソフトウェア開発論を大上段に振りかざさなくても、さりげなく HCP チャートを使うだけで自然に品質の良いソフトウェアを

得ます。本書では、ソフトウェア開発の各工程で、HCP チャートをどのように使っていくかを見ていきます。

なお、HCP チャートは、その後昭和 63 年度から平成 3 年度まで日本規格協会の INSTAC-STD-WG3 (図記号の標準化) においても取り上げられ、NTT ソフトウェア研究所などにより改良が加えられ、現在に至っています。

本書では、執筆者の経験を踏まえ、HCP チャートを使用した実戦的なソフトウェア開発方法を紹介します。このため、HCP チャートの使用方法および図示方法が、オリジナルと多少異なる部分がありますが、ご了承下さい。オリジナルの HCP チャートについては、HCP チャートの開発者によって書かれた優れたテキスト [1][2]をご参照下さい。

なお、本書で例として掲載されている HCP チャートは、全て HCP チャートエディタ *nnHEADWAY/P for Windows*¹を用いて作成しました。

1995 年 執筆者

一部加筆訂正を行い第 2 版としました。

1997 年 執筆者

¹ 株式会社デンソークリエイト製 問い合わせ先 E-mail:info@dcinc.co.jp tel. 052-229-1192

(中略)

HCP チャートは、フローチャートしか見たことのない方にとっては、かなり変わって見えるかもしれませんが。しかし、HCP チャートは、フローチャートよりはるかに使いやすいチャートです。その理由を1つ1つ見ていきます。

処理説明文の書き方

HCP チャートでは、説明文は処理記号（ 印など）の右側に自由に記述します。一方、フローチャートでは、処理記号（長方形など）の中に処理の説明を記述します。ところがそうしますと、処理説明文が長くなると、処理を表現する長方形も同じように長く書かなければなりません。

HCP チャートでは説明文を処理記号の制約なく長く書くことができます。フローチャートのように、いちいち処理記号の大きさに頭を悩ませる必要はありません。

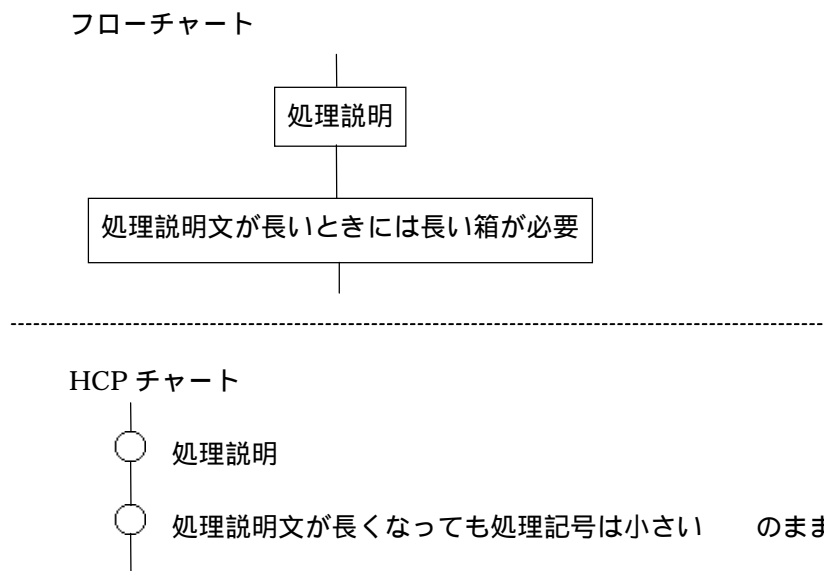


図1 処理の説明文の書き方

処理の流れ

HCP チャートは必ず上から下へ処理が進みます。これは、とても分かりやすい表記です。ところが、フローチャートでは、下から上へ線が伸びることもあります。そうしますと、行ったり来たりしてチャートを読む必要がありますので、非常に読みづらく、分かりにくくなります。特に、上向きの線が長くなり、また、その数が増えてきますと、極端にチャートが読みにくくなります。

(中略)

繰り返しのための処理記号

構造化プログラミングチャートである HCP チャートは、構造化プログラミングの1つの特徴である「繰り返し」を特徴的な記号を用いて簡潔に良く分かるように表現できます。

繰り返し処理

④ 処理ラベル 処理説明

この記号は、繰り返し制御構造を表現するために用います。

処理説明には、「この繰り返し処理により何が行われるか」を書きます。例えば、「最小値を求める」というように、繰り返し処理全体において行われる処理の内容を書きます。「...まで繰り返す」という表現は使いません。チャートの記号を見れば、そのようなことはすぐに分かります。わざわざ文字で説明しても何の情報量も増えません。

この記号のすぐ右側に、レベル開始記号を書きます。つまり、上から来た処理は、ここでレベル開始記号に従って右側に処理の向きを変えます。そして、後ほど述べる例外条件および飛び越し条件が成立するまで、永遠にこの繰り返し処理を繰り返します。

繰り返し処理記号 ④の丸くなった矢印が、繰り返しを端的に表現しています。HCP チャートでは、このような特徴的な記号を使うことで、選択構造の表現を際立たせています。また、フローチャートのように上向きの矢印線を用いませので、かなり見やすくなっています。



図2 繰り返しの例

(中略)

第2節 外部仕様設計

外部仕様設計の工程では、運用や操作方法などを中心とした利用者の立場で、要求仕様に基づいてシステムの働きを決めます。

要求仕様を理解するためには、仕様の全体像をポンチ絵として書くなどの工夫をします。その上で、ユーザの立場に立って、運用や操作方法を定めます。もちろん、この段階では、対象システムに関する幅広い知識が必要になります。例えば、給料計算システムの外部仕様を設計する段階では、残業や勤怠などの会社の勤務ルールや銀行振り込みに関する知識が必要になります。

この工程では、HCP チャートはシステムの運用手順の設計やユーザインタフェースのメニュー設計を行うときに、広く利用できます。

例えば、システム運用において、ある条件が成立するまである処理を繰り返すという場面がよくあります。これは、HCP チャートの繰り返しと条件分岐により、正確に分かりやすく表現できます。

また、ユーザインタフェースでは階層メニューを作る必要があることがよくあります。メニューの階層構造は、HCP チャートを使うと正確に分かりやすく表現できます。

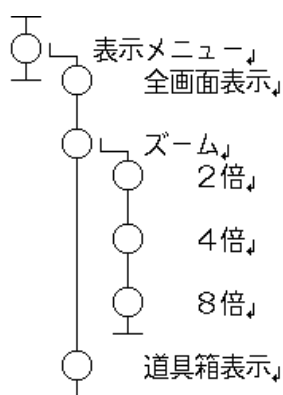


図3 メニュー設計の例

このように、外部仕様を作成する段階においても、HCP チャートは有効な道具として利用できます。

(以下省略)