

## HCP チャートエディタの実現と評価実験

中神明 塩見 彰睦 竹田 尚彦 河合 和久 大岩 元

豊橋技術科学大学

慶應義塾大学

PAN/HCP は、HCP チャートの編集を計算機上で行なうツールである。本稿では、PAN/HCP のパーソナル・コンピュータからエンジニアリング・ワークステーションへの移植について述べる。移植にあたっては、開発言語を C 言語から C++ に変更した。また多機種への対応を考え、ウィンドウシステムの 1 つである X-Window System 上で動作するアプリケーションとして開発を行なった。移植の際には、いくつかの機能拡張を行なった。さらに、機能評価を目的として評価実験を行なった。その結果から、自由に記述できることの有効性、チェック機能の問題点について述べる。

### IMPLEMENTATION AND EXPERIMENTATION OF A HCP CHART EDITOR

Akira Nakagami Akichika Shiomi Naohiko Takeda Kazuhisa Kawai Hajime Ohiwa

Toyohashi University of Technology

Keio University

PAN/HCP is a tool to edit HCP charts on a computer. We describe the transportation of PAN/HCP from Personal Computer to Engineering Work Station. On the transportation, we implemented some additional functions, changed the implementation language from C to C++, and developed as an application of X-Window System that is one of the most popular window system for corresponding to portability. Then, we experimented for the evaluation of PAN/HCP functions. From the result of this, the effect of the free description of HCP charts and some problems in the chart checking function are identified.

## 1 はじめに

プログラムの機能仕様からプログラムの内部的な設計(詳細設計)をするときに、プログラムのアルゴリズムを開発言語に依存しない形で表現すると、さまざまな言語を利用している多くの人に共通して理解できるドキュメントとなるとともに、開発言語の変更にも対処し易くなるという効果がある。アルゴリズムを開発言語に依存しない形で表現する方法として、さまざまなプログラム設計図法が提案され、多くの人に利用されている。

我々は、上流工程の詳細機能設計以降の設計工程を支援する構造化チャートとして、HCPチャート<sup>[1]</sup>に注目した<sup>[2]</sup>。HCPチャートは(1)トップダウンな設計作業に向いており、(2)目的(What)と手段(How)の関係をもとに自然言語を用いて記述する。さらに、(3)実際のプログラム構造とも対応している、という特徴がある。このため、HCPチャートは、プログラミング経験のない人でも理解し易いドキュメントとなる。

設計作業は試行錯誤的に行なわれることが多く、図式の追加・修正・削除の必要が頻繁に起こる。しかし、紙の上に記述したチャートは修正が容易ではないため、設計作業よりもむしろ図の修正のために時間がかかってしまうことも少なくない。そこで我々は、図の修正といった編集作業を軽減し、チャートの保管を容易にするために、計算機上でHCPチャートの編集を行なうツールであるHCPチャートエディタ“PAN/HCP”をパーソナルコンピュータ(以下パソコンとする)上に実現している<sup>[3]</sup>。

しかし、パソコン上のPAN/HCPはメモリの制約により、扱うことのできるデータ量に制限があるとともに、機能の拡張も不可能となっていた。また、最近ではエンジニアリング・ワークステーション(以下EWS)が普及し、大学や企業において広く利用されるようになってきている。そこで、PAN/HCPの今後の機能拡張と、EWS上でのソフトウェア設計環境を整備するために、EWS上で動作するPAN/HCP(以下EWS版PAN/HCP)を実現した。そして、その機能評価をするために、EWS版PAN/HCPを用いて実際の設計作業をする評価実験を行ない、その結果からPAN/HCPの機能について考察を行なった。

2章でPAN/HCPの特徴について述べる。3章では、EWS版PAN/HCPの設計と実現方法について述べる。そして4章では、EWS版PAN/HCP

を用いた評価実験の課題や実験方法、5章では実験結果から得られたデータに基づくPAN/HCPの機能評価について述べる。

## 2 HCPチャートエディタ“PAN/HCP”

### 2.1 PAN/HCPの特徴

PAN/HCPは、HCPチャートの記述を計算機上で支援するHCPチャートエディタである。他にHCPチャートエディタとして代表的なものにHDシステム<sup>[5]</sup>がある。HDシステムは、NTTソフトウェア研究所で開発されたHCPチャートエディタであり、C言語やCOBOLなどのプログラムを自動生成する機能を持つ。

PAN/HCPは、単なる清書作業やプログラムの自動生成のためのツールではなく、詳細設計工程における試行錯誤的な詳細設計を快適に行なうことを支援する、ということに重点を置いている。

他のHCPチャートエディタではトップダウンにしか記述できないという制限を与えているものがある。それに対してPAN/HCPではトップダウンにこだわらない記述を許している。

本来、HCPチャートはトップダウンでの記述が基本であり、トップダウンで設計していくことは概要から詳細へと全体の構造を考えていく上で良いことである。しかし、アルゴリズムやデータ構造を設計する時には、おおよその流れはトップダウン思考で設計することができるものの、今までの経験などから、時には部分的にボトムアップで考えがまとまることもある。例えば、HCPチャートを紙の上に書いているときには、下階層のものでも先に記述しておくことができた。HCPチャートエディタに、トップダウンでしか記述できないという制限を設けると、この下階層から思いついたものを記述できなくなってしまう。

PAN/HCPでは、書きたいところ、また書けるところから自由に記述していく方法を採用しているため、設計者は思い浮かんだ処理をその都度チャートに記述しておけば、その内容を覚えておく必要がなく、それだけ設計に集中できるという効果が生まれる。

また、PAN/HCPでは、トップダウンでしか記述できないという制限をなくしたために、不完全なチャートや記述誤りを含むチャートができてしまうことがわかった<sup>[3]</sup>。不完全チャートなチャートがそのままシステム設計以降の工程に渡されると、後で

手戻りが必要となるなど、効率の良いプログラム作成の妨げとなる。そればかりか、システム設計自体の信頼性に関わる問題となる。

そこで、チャートの記述誤りと記述忘れを発見するために、PAN/HCPはチェック機能を持っている。この機能により、自由に記述できることによるチャートの記述忘れや、構造上の誤りを発見することができる。

また、このチェック機能は編集時に常時チャートをチェックし監視するのではなく、利用者の指示によってチェックをするように実現されている。これは自由に記述できるという利点を生かすためである。

## 2.2 PAN/HCPの表現形式

HCPチャートの記述には広い記述領域が必要となる。紙の上にHCPチャートを記述する場合には、少なくともA3大程度の用紙を用いなければ設計作業を効率的に行なうことはできない。それに対して、現在一般に利用されているディスプレイには、A3用紙ほどの広い記述領域を表現するだけの解像度はない。

そこで、PAN/HCPでは、仮想的な記述領域を用意し、図式要素はこの上に記述する方式をとっている。この仮想的な記述領域を机と呼ぶ。図式要素とは図を構成する記述単位であり、HCPチャートでは処理やデータの記号、制御線、データと処理の関係を表す線、などが図式要素となる。

また、PAN/HCPでは、図全体を一度に画面に表示できないために、その一部のみをローカル画面に詳細に表示している。そして、ローカル画面からマウスカーソルがはみ出るようにマウスを操作した時に、ローカル画面からはみ出た移動距離にあわせて画面をスクロールさせる方法を採用している。この方法をパニングと呼ぶ。また、図全体を一覧するために、ローカル画面とは別に図全体の概略図をユニバーサル画面に表示している。ユニバーサル画面には、現在ローカル画面がどの部分を表示しているかという範囲枠も表示される。

机とローカル画面、ユニバーサル画面の関係を、図1に示す。

## 3 EWS版PAN/HCPの設計と実現

PAN/HCPをEWS上を実現し、実現の際にはいくつかの機能拡張を行なった。この章では、移植

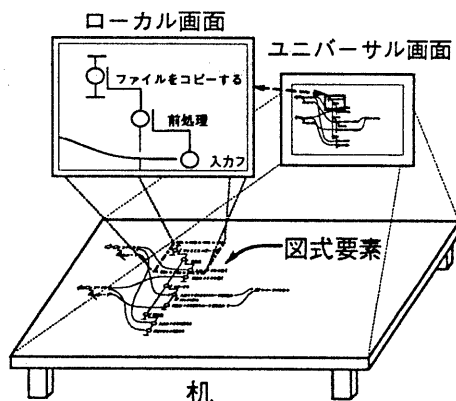


図1 PAN/HCPの表現形式

の目的と、移植の際に拡張した機能について述べる。

### 3.1 EWS上への移植の目的

パソコン版PAN/HCPは、HCPチャートの編集に必要な機能を持っている。しかし、パソコンのメモリの制約により、一枚のチャートに記述できる図式要素数や線の長さに制限があるとともに、機能の拡張も不可能となっていた。一般的に、EWSはパソコンと比べて、使用可能なメモリが多く、処理能力や表示能力も優れている。EWS上へとPAN/HCPを移植することにより、PAN/HCPの機能拡張が可能となる。

また、最近では大学や企業における研究・開発を行なう計算機環境はパソコンからEWSへと移行しつつある。EWS上でのソフトウェア開発環境を整備するには、EWS上で動作するPAN/HCPを実現する必要がある。

そこで、PAN/HCPの機能拡張と、EWS上での開発環境を整備することを目的として、EWS版PAN/HCPを実現することにした。

EWS版PAN/HCPの実現には、EWS上で広く用いられているX-Window Systemを採用し、開発はC++を用いて行なった。

今回EWS上を実現したPAN/HCPの実行画面を図2に示す。

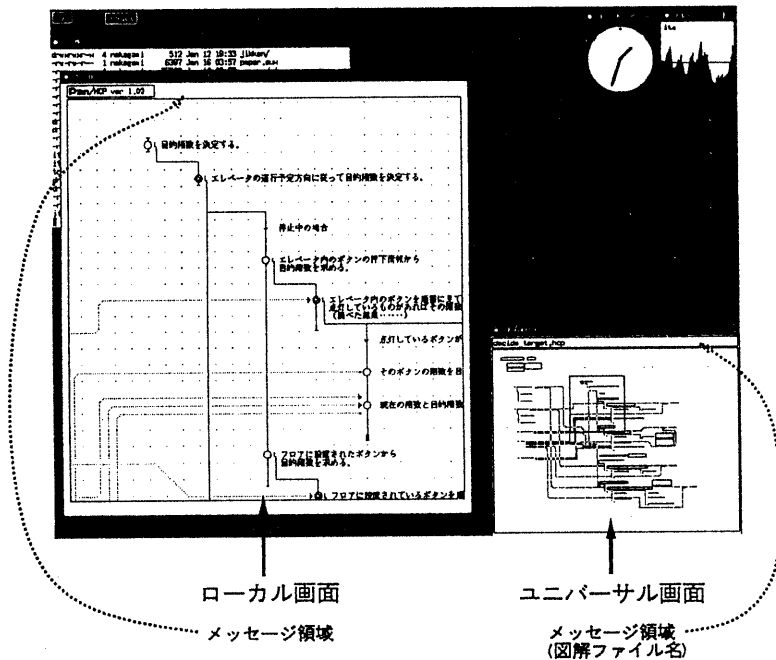


図 2 EWS 版 PAN/HCP の実行画面

### 3.2 EWS 版 PAN/HCP の拡張機能

EWS 版 PAN/HCP は、基本的にはパソコン版 PAN/HCP と同じ操作方法・機能をそのまま実現をすることとした。これは、パソコン版 PAN/HCP の試用実験<sup>[3]</sup>の結果から、基本的な操作については良いという結果が得られているからである。

しかし、パソコンと EWS の違いから、いくつか修正が必要となる機能があることがわかった。また、パソコン版 PAN/HCP ではパソコンの機能、メモリの制限から実現できなかった機能についても、新たに機能拡張を検討をした。

以下に、パソコン版からの変更点を示す。

#### データと処理の関係の表現方法を変更

パソコン版 PAN/HCP ではデータと処理の関係を、1本の線ではなく、その両端に流れの向きと線の通し番号を表示することで実現していた。この記述方法を用いた場合、データと処理との関係を表す線が交錯せずに、HCP チャート全体は非常にすっきりしたものとなる。しかし、線の本数が増える

と、分断した線の両端を識別することが困難となり、HCP チャートの理解を妨げる原因となる。そこで、本来の HCP チャートの記述方法と同じ、1本の線で表すようにした。

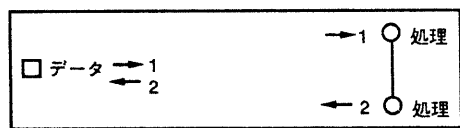
#### メニューの見直し

パソコン版 PAN/HCP では、編集モードの変更と機能の実行のために、1つのメニューを用意していた。また、それとは別に、ファンクションキーから選択を行なう機能もあった。EWS 版 PAN/HCP では、選択方法を統一するために、ファンクションキーを用いた機能選択を廃止し、メニューから選択を行なうことにした。また、モード変更と機能実行は別のメニューとして実現することとした。

#### ショートカット・キーの導入

マウスを用いたメニュー選択方式は、キーボード上のキーと機能との対応関係を覚える必要がないため、操作方法の習得が容易である。しかしその反面、アプリケーションの操作に慣れてくると、メ

### パソコン版PAN/HCP



### EWS版PAN/HCP

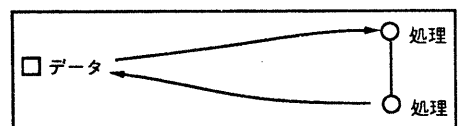


図3 データと処理の関係の表現方法

ニューからの選択は煩わしく感じることもある。そこで、メニューを用いずにキーボードから直接機能を選択する機能を追加した。

#### ログ取り・ログ再生機能の追加

ログ取り機能とは、編集履歴を保存する機能である。また、ログ再生機能とは、ログ取り機能により得られたログデータから、ログ取りを行なった時の編集操作を再現する機能である。

ログデータからは、図式要素数の時間的変化や編集操作回数、マウスの総移動距離などさまざまなデータを得ることができる。そして、このデータから、PAN/HCPの機能評価を客観的に行なうことが可能となると考えた。

また、我々は、PAN/HCPを教育に利用することを考えている。ログ取り・ログ再生機能により、(1)教材の1つとして、良い設計者の設計作業を初心者に見せる、(2)初心者の設計の過程を指導者が見る、ということがVTRなどの特別な装置を必要とせずに可能となる。

ログ取り・ログ再生機能を実現するためには、ログデータの入出力や、ログデータのファイル出力用フォーマットを管理する機能が必要となる。そこで、これらの機能を実現するクラスLogManagerを作成した。

また、ファイルにデータを保存する形式としては、テキスト形式を採用した。テキスト形式を用いることにより、ファイルの内容をテキストエディタなどを用いることにより、PAN/HCPを用いることなく編集操作の流れを簡単に追うことが可能と

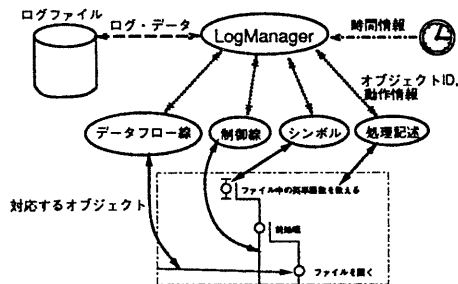


図4 LogManagerの役割

なっている。また、awkやsedといった文書処理言語を用いて、簡単にログデータを解析することが可能である。

## 4 PAN/HCPの評価実験

### 4.1 実験目的

今回EWS上で実現したPAN/HCPを用いて、実際の詳細設計を行ない、その設計過程から客観的なデータを得る。そして、そのデータに基づいてPAN/HCPの機能を評価する。

### 4.2 実験課題

実験課題としては、「ソフトウェアの仕様記述と設計に関する第4回国際ワークショップの課題」の中のProblem 4: LIFTを選んだ。その内容はおおむね以下の通りである。

m階の高さのビルに備え付けられている1台のエレベータを、以下の制約に従って運転するためのロジックを設計する。

各階には上下方向を表す1組ボタンが設置され、エレベータはこのボタンの要求に答えなければならない。全ての階には等しく優先権が与えられており、エレベータは最終的に各階からの全ての要求に応じなければならない。また、エレベータには各階に対応した1セットのボタンがある。このボタンが押されると、対応する階へエレベータを運転する。エレベータは、運行方向に従ってエレベータ内の

表 1 実験結果

	被験者 A	被験者 B	被験者 C	被験者 D
PAN/HCP 使用時間	5 時間 4 分	7 時間 49 分	5 時間 32 分	5 時間 40 分
提出用チャート (枚)	2	6	4	4
チャートファイルサイズ (KB)	41	36	61	60
ログファイル数	5	6	6	4
ログデータ数	226054	367044	455433	184946
途中で印刷をしたチャート	0	0	4	5
手書きのメモ	0	0	0	1

要求に順序よく応じ、最終的にはすべての要求に応じなければならない。

Problem 4:LIFT では  $n$  台のエレベータとなっている部分を、1 台のエレベータと問題を一部変更している。これは、 $n$  台の場合は問題が複雑になり設計が容易ではないと考えたからである。

### 4.3 被験者

被験者は HCP チャートに関する知識と記述の経験などから、被験者 A～被験者 D の合計 4 名を選定した。

被験者 A と被験者 B は計算機の使用経験を十分に持っており、初級プログラマとしての能力と経験を持つ。普段はプログラム設計図法を用いた詳細設計はほとんどせず、HCP チャートの記述経験はない。また、PAN/HCP の使用経験はない。

被験者 C と被験者 D も計算機の使用経験を十分に持っており、上級プログラマとしての能力と経験を持つ。また、被験者 D は SE としての経験も持つ。2 人とも、アルゴリズムが難しい場合のみ HCP チャートを用いて詳細設計をしている。また、パソコン版 PAN/HCP は使い慣れている。

### 4.4 実験方法

実験は以下の条件に従って行なうように指示した。

- ・ 設計作業、特に HCP チャートの記述は EWS 版 PAN/HCP 上で行う。
- ・ 記述するチャートの枚数や設計時間は自由である。
- ・ 設計作業のログを PAN/HCP のログ取り機能により採取する。

- ・ 被験者 B のみは、トップダウンでしかチャートの記述をしてはいけない。

被験者 B のみに、トップダウンで設計をしなければならないという条件を与えた理由は、PAN/HCP の特徴の一つである自由に記述できることの効果を調べるためである。

## 5 評価実験の結果と考察

### 5.1 各被験者の作成物と設計時間

PAN/HCP を用いて設計作業を行なった設計時間と結果として得られたものを表 1 に示す。

各被験者とも、設計作業を数回に分けて行なっている。PAN/HCP 使用時間は設計時間の総計である。

提出用チャートは、今回の実験で作成した HCP チャートである。HCP チャートのモジュール化を利用することにより、全員複数枚のチャートを提出している。また、チャートは PAN/HCP で取り扱う図解データファイルのままの形で提出させた。

途中で印刷をしたチャートとは、PAN/HCP 上で記述中の HCP チャートを PAN/HCP の印刷機能によって印刷したものである。それらにはいずれも修正を意味する書き込みがなされていた。

手書きのメモは PAN/HCP を用いて設計を行なう前に書いたものであり、与えられた問題に関するデータ構造、エレベータの持つボタンからのデータ受けとり方法、アルゴリズムの基本方針などさまざまなものが書かれている。

### 5.2 提出されたチャートの比較

各被験者の最終提出用の HCP チャートから、書かれているそれぞれの図式要素数を調べた結果、HCP チャートの初心者である被験者 A,B について

表 2 提出用チャートの制御構造の深さ

	被験者 A	被験者 B	被験者 C	被験者 D
最大	14	4	5	7
最小	6	3	3	5
平均	10.0	3.7	4.5	5.8

は、データのシンボル数とデータと処理との関係の記述が非常に少ないことがめだつた。これは、普段の設計作業の際に、アルゴリズムには注意しながら設計するが、データ構造の設計やデータの流れの把握をおろそかにしているからである、と考える。データ構造とデータの流れについても、しっかりと設計をしなければ HCP チャートの特徴を生かすことはできない。初心者には HCP チャートの記述を指導するにあたっては、処理の流れと同様に、データ構造とデータと処理の関係をきちんと書くことを説明することが必要である。

また、各被験者の記述したチャートにおける段階の詳細化の段数比較を表 2 に示す。

被験者 B,C,D はチャートの段数にあまり差がなく、また段数も極端に多くはない。それに対して、被験者 A は最大 14 段目の階層を持つチャートを記述している。1 枚のチャートにあまり多くの記述をすると、チャートが非常に読みにくくなってしまふ。これは、適切なモジュール化によって解決できる問題である。

処理構造の段数が多くなる記述を禁止するのは容易である。しかし、試行錯誤で設計作業を行っている過程において記述制限を与えることは、PAN/HCP の自由に記述できるという特徴を損なう恐れがある、と考えている。そこで、記述自体は可能であるが、チェック機能実行時に段数が多いチャートについては警告をするという方式を検討している。

### 5.3 自由な記述を許すことによる効果

各被験者が設計中に図式要素をどれだけ発生したか、そのうちでどれだけ削除をしたかを表 3 に示す。カードは、設計メモやチャートの補足説明を記述することに用いる図式要素である。また、データフロー線とはデータと処理の関係を表す線である。

残存率は、発生した図式要素がどれだけの割合

表 3 図式要素の発生数と消去数

被験者	図式要素	発生数	消去数	残存率 (%)
被験者 A	シンボル	251	30	88.0
	カード	4	1	75.0
	制御線	276	59	78.6
	データフロー線	16	8	50.0
	計	547	98	82.1
被験者 B	シンボル	357	259	27.5
	カード	83	30	63.9
	制御線	288	203	29.5
	データフロー線	40	30	25.0
	計	768	522	32.0
被験者 C	シンボル	262	86	67.2
	カード	35	7	80.0
	制御線	248	102	58.9
	データフロー線	85	12	85.9
	計	630	207	67.1
被験者 D	シンボル	225	28	87.6
	カード	7	0	100.0
	制御線	194	30	84.5
	データフロー線	42	3	92.9
	計	468	61	87.0

でチャート完成時に残っているかを表す数であり、

$$\text{残存率} = \frac{\text{発生数} - \text{消去数}}{\text{発生数}} \times 100 \quad (\%)$$

により求める。

また、設計作業中におけるシンボル数の時間的変化を表したグラフを図 5 に示す。

表 3 から、被験者 B におけるカード以外の図式要素の残存率が極端に低い (25% ~ 30%) ことがわかる。また被験者 B は、図 5 において、設計時間が 200 分 ~ 300 分の段階で、シンボル数が増加・減少を繰り返していることがわかる。

トップダウンでしか記述できないと、一度処理構造を記述しても、処理階層の中間部分に処理を追加したい場合は、それより下のレベルについての記述を削除しなければならない。被験者 B はトップダウンでしか記述できないという制限により、残存率が低くなっている、と考える。

今回の実験の結果から、トップダウンでしか記述できない HCP チャートエディタを用いて設計作業を行なうと、修正のための記述・削除が頻繁に起こり、設計作業に集中できなくなる場合があることが指摘できる。

また、表 3 において、被験者 B 以外の被験者に

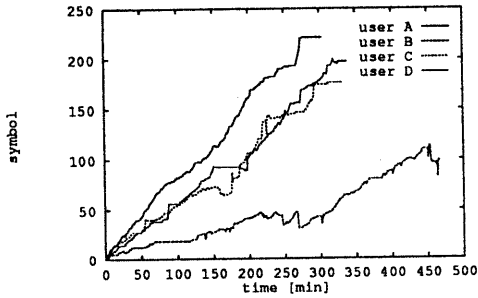


図5 シンボル数の時間的変化

については、残存率は50%から90%弱となっている。PAN/HCPを単なる清書のために用いているのならば、もっと残存率は高くなり、100%にはほぼ等しくなるはずである。このことから、全員試行錯誤の設計作業をPAN/HCP上で行なっていたことが分かる。

#### 5.4 チェック機能の評価

PAN/HCPの持つチェック機能をどれだけ設計が進んだ段階で行なっているか、また、どれだけのチャートの記述誤り(記述忘れ)を発見したかを調べた。その結果、被験者Dのみはチェックを早い段階からしていることがわかった。それにもかかわらず、全員の提出したチャートをあらためてPAN/HCPのチェック機能によりチェックした結果、被験者Dが提出したチャートの1枚から、記述誤りを発見した。

PAN/HCPのチェック機能の本来の目的は、利用者が気がつかなかったチャートの不完全な部分を見つけ、利用者に知らせることにある。それに対して、被験者Dは、チェック機能をチャート内の処理構造が分断している箇所の一覧や、処理構造が分断している箇所へとジャンプする目的で利用していることが分かった。PAN/HCPのチェック機能では、不完全な部分を見つけた場合には、瞬時に画面がパニングするので、このような利用方法も可能である。しかし、普段チェック機能をチャートに記述誤りや記述忘れがないかを見出す目的以外に用いていたため、結局誤りを含んだチャートが提出される結果となった。

そこで、新たに処理構造が分断している箇所の一覧や処理構造が分断している部分へジャンプする機能は別に設けて、このような目的でチェック機能を実行しないように指導する必要がある。

#### 6 まとめ

本研究では、パソコン上で実現されているHCPチャートエディタPAN/HCPを、機能拡張と計算機環境の変化に対応することを目的とし、EWS上に実現した。実現の際には、データと処理との関係の表現方法の変更や、ログ取り・ログ再生機能の追加を行なった。

また、今回EWS上で実現したPAN/HCPを用いて、実際の詳細設計を行ない、その設計過程から得られるデータを用いて、EWS版PAN/HCPの機能について評価を行なった。設計過程の記録方法としては、今回追加したログ取り機能を用いた。

#### 参考文献

- [1] 花田 收悦, プログラム設計図法, 企画センター, 1983.
- [2] 塩見 彰陸, 河合 和久, 竹田 尚彦, 大岩 元, ソフトウェアの基本設計と図式エディタ PAN, 情報処理学会「CASE環境」シンポジウム, pp.41-48, 1989.
- [3] 塩見 彰陸, 竹田 尚彦, 河合 和久, 大岩元, HCPチャートエディタ PAN/HCP, 情報処理学会論文誌 第33巻 第2号, pp.183-194, 1992.
- [4] 富田 理, 重盛 勝, 竹田 尚彦, 河合 和久, 大岩元, カード操作ツールのC++によるUNIXへの移植, 第18回 jus UNIX シンポジウム論文集, pp.80-96, 1991.
- [5] NTTソフトウェア研究所, HDシステムマニュアル, 1986.
- [6] 馬場 勇, ソフトウェア開発の実践技法, 技術評論社, 1989.
- [7] 河村 一樹, ソフトウェア工学入門, 啓学出版, 1987.