

HCP チャート清書ソフトウェア「HCP Viewer」の開発

松澤 芳昭[†] 杉浦 学[†] 大岩 元[‡]

[†]慶應義塾大学 政策メディア研究科 [‡]慶應義塾大学 環境情報学部

macchan@crew.sfc.keio.ac.jp

概要

HCP チャートによる設計教育の支援ツール「HCP Viewer」を開発した。初学者でも容易にチャートを作成でき、また頻繁なレビュー/改訂のプロセスにも対応できるよう、編集しやすく、創造的な作業に集中できる環境を目標とした。HCP Viewer は、WYSIWYG 方式のエディタではなく、コマンドとインデントによってマークアップされたテキスト文書を清書するという方式を採用した。この設計判断は、HCP チャート上に配置された処理要素の本質的な情報が、二次元空間上の位置ではなく、階層構造上の位置によって表現されるという特徴を踏まえている。評価実験と運用経験を通して、処理構造の試行錯誤的な編集がしやすく、初学者にも使いやすい環境であることが確認できた。データの記述に関しては、既存のエディタと同等の編集性能を確認できた。

1. はじめに

筆者は、プログラム設計の基礎教育として、HCP チャート (Hierarchical and Compact Description Chart) [1][2]を用いた教育実践を行っている[3][4]。

HCP チャートは、日本電信電話公社横須賀電気通信研究所で1980年代初頭に開発された設計図法で、プログラムを処理の目的(What)とその実現手段(How)の関係に注目し、階層構造にまとめて表現するという特徴をもつ(例: 図1)。人間が、複雑なシステムを把握するために、階層構造を利用して整理する考え方は、開発手法の主流がオブジェクト指向になった今日でもなお一般的である。

上記の設計教育では、作成された HCP チャートに対して第三者がレビューをし、作成者が改訂を行う。レビュー/改訂のプロセスは頻繁に、反復的に行われるため、これを効率よく行なうためにはコンピュータの支援が欠かせない。

HCP チャートの作成支援システムは、塩見らによって「PAN/HCP」[5]が提唱された。近年では、シェアウェアの「HCP CHART EDITOR」[6]や、デンソークリエイイト社「nnHEADWAY」[7]などのシステムが登場して

いる。これらのシステムは WYSIWYG (What You See Is What You Get) 方式という共通の特徴を持つ。

WYSIWYG 方式のエディタは、清書された画面をそのまま編集できるという利点を持つ。しかし、HCP チャートの作成時においては、記法の制約のため、トップダウンとボトムアップの思考が交錯する試行錯誤的な作業を阻害するという問題点がある[5]。その問題を改善した「PAN/HCP」では、逆に清書作業によって創造的な作業が阻害されてしまうことが指摘されている。

WYSIWYG 方式のエディタは特有の操作を要求する。それゆえ、操作方法の習得プロセスも初学者にとって創造的な作業を阻害する、といった教育上の欠点もある。

上記の理由から、特に教育現場ではこれらのツールを学習者に薦めづらい。

こうした背景から、筆者はマークアップ言語方式で初学者でも容易に HCP チャートを作成、改訂することができる支援ツール「HCP Viewer」を開発した。本稿では、HCP Viewer の設計思想、概要、および性能試験の結果と実際の利用経験に基づく評価を述べる。

“HCP Viewer” - An HCP Chart Editing System
Yoshiaki Matsuzawa[†], Manabu Sugiura[†] and Hajime Ohiwa[‡]
[†] Graduate School Media and Governance, Keio University
[‡] Faculty of Environmental Information, Keio University

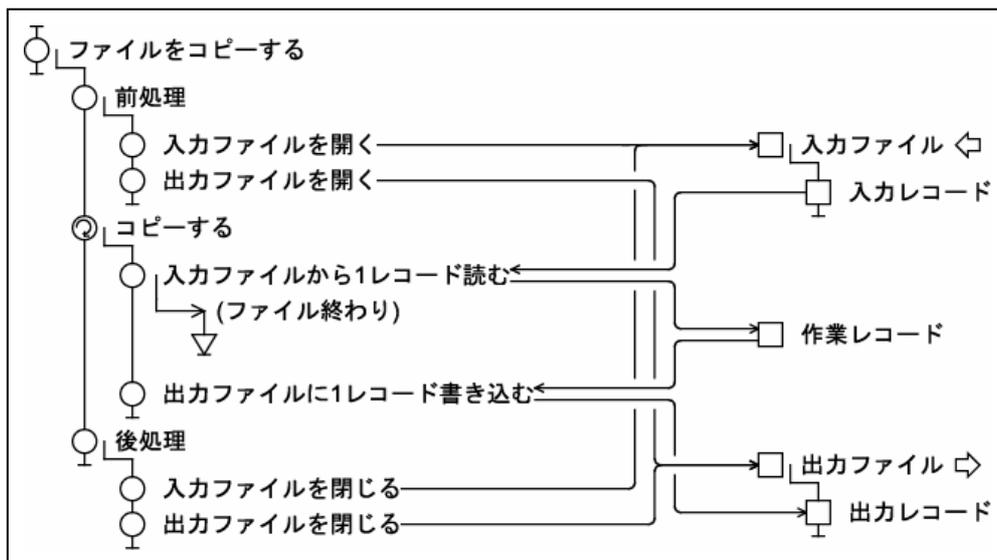


図 1 HCP チャート例 (ファイルをコピーする)

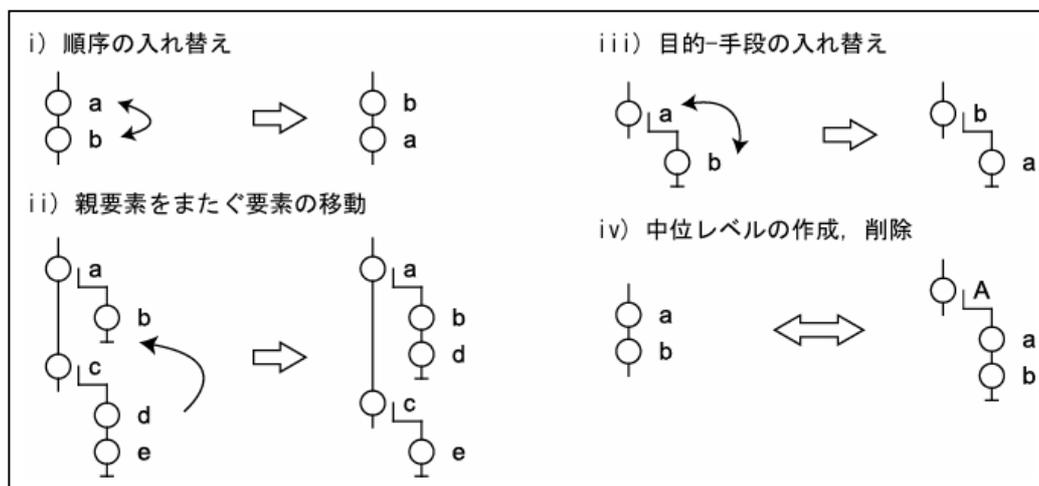


図 2 改訂パターン

2. HCP Viewer の設計思想

2.1. HCP チャートの作成プロセス

塩見は、HCP チャートがトップダウンとボトムアップの思考が交錯する試行錯誤的なプロセスにより作成されることを指摘した[5]。これについては、筆者も同様の考えである。

筆者の教育現場では、一旦完成したチャートにレビュー/改訂が繰り返される。この際の改訂のパターンは i) 順序の入れ替え, ii) 親要素をまたぐ要素の移動, iii) 目的-手段の入れ替え, iv) 中位レベルの作成, 削除が多く (図 2), 大規模な改訂の場合は i) ~ iv) が複合的に行われることもある。これは、トップダウンともボトムアップとも捉えられない作業である。

2.2. 二次元配置によるボトムアップ設計支援の是非

塩見の主張は、計算機の仮想ディスプレイ上に設けられた大きな二次元空間に自由に要素を配置できるようにし、後から制御線¹を結ぶ、という方法により上記の試行錯誤のプロセスが支援される[5]というものである。筆者はこの考えに疑問を持つ。要素の位置を決める思考が設計の本質的な思考を阻害する可能性があるという理由からである。

HCP チャートによって表現された各処理要素は階層構造上での位置のみによって役割が規定されるのであって、二次元空間の位置は

¹ 例えば、図1で「前処理」と「コピーする」を結んでいる線

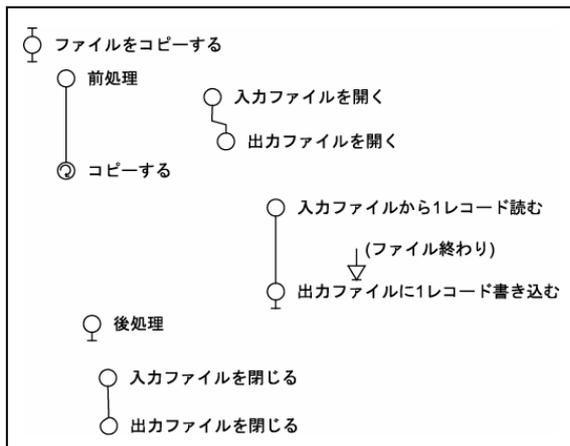


図 3 二次元空間的な配置で表現した PAN/HCP

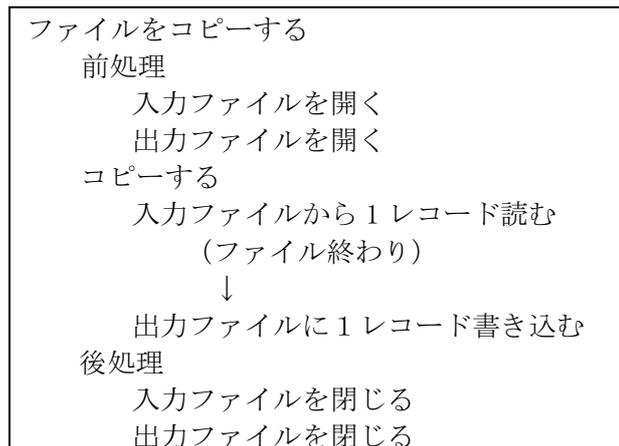


図 4 インデントで表現した テキスト HCP

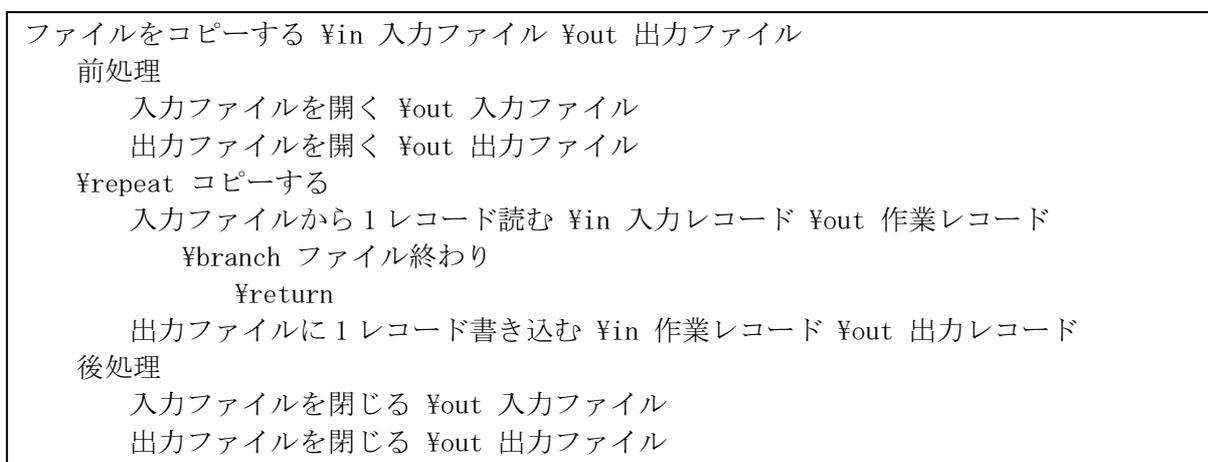


図 5 HCP Viewer が処理できるマークアップテキストの記述例 (図 1 を出力する)

その情報をもたない。例えば、図 3 は PAN/HCP を用いて図 1 の HCP チャートを作成する際の途中段階を図示したものであるが、各処理要素がどのレベルに属するのかが明らかではなく、全体の構造はまだ把握しづらい。逆に、図 4 では同様の状況をインデント付きのテキストで表現しているが、これでも階層構造の情報は保持されており、全体像が把握できる。(尚、ここでは配置について議論しているため処理の種類は考慮していない)

ワードプロセッサの出現によって「物書き」自体のプロセスが変化したように[9]、計算機では、チャート作成のプロセスも変化する可能性がある。手書きの時は階層構造の作成を補助していた二次元配置は自由度が高すぎるため、階層構造の直接的な編集ができる計算機上では必ずしも最善のプロセスとはいえない。

2.3. 制御線の機能

PAN/HCP をはじめとする WYSIWYG 方式のエディタでは、制御線を引いて一旦完成した後の改訂も考慮されていない。

制御線の主な機能は処理のレベルが同じであることを示すことである。これは、図 1 と図 4 を比較して制御線の機能を分析することにより分る。従って、インデントによって階層の位置が明らかにされているのならば、制御線の有無によって情報量は変化しない¹。

制御線のもうひとつの機能として、清書した際にチャートの読みやすさを向上させる機能がある。しかし、インデントで表現されたチャートを編集する際には、編集作業を混乱

¹ 制御線は、手書きでチャートを作成する際、空間的余裕がない場所に処理を挿入するとき役立つ記法として設計されたものと推測される

させる要因となる¹。

3. HCP Viewer の概要

3.1. マークアップ言語方式による実現

前述の設計思想に従って、筆者は HCP チャート清書ツール「HCP Viewer」を開発した。HCP Viewer は WYSIWYG 方式のエディタではなく、コマンドとインデントによりマークアップされたテキストを清書する方式(マークアップ言語方式)を採用した。この理由を以下に示す。

i) 使い慣れたエディタを使用できる

編集するエディタを問わないので、使い慣れたエディタで、使い慣れた編集操作により編集ができる。(設計学習者は少なくとも小規模のプログラミングを経験しており、プログラミング経験者ならばエディタの操作は熟知している)

ii) ソースの可読性を高く保てる

一般的に、マークアップ言語方式の場合、マークアップされたソースの可読性が低いという欠点がある。しかし、前述のように、HCP チャートの本質的な情報は、空間的な配置ではなく、階層構造によって表現される。それゆえ、インデントによるマークアップソースは、清書のイメージを完全に把握できるほど可読性を高く保つことができる。

iii) 編集がしやすい

制御線のないテキストを編集するので、特に 2.1 節で示した改訂作業がしやすく、創造的な作業に集中できる。

iv) フォワードエンジニアリングが容易

HCP チャートの処理記述は、実装時にソースコードのコメントとして活用したい。これを WYSIWYG 方式のエディタで実現しようとする特殊なフォワードエンジニアリング機能が必要である。マークアップ言語のソースはテキストなので、コメントとしての再利用が容易である。インデントもそのまま利用できる。

例えば、HCP Viewer を用いて図 1 のチャートを清書するソースは図 4 のようになる。

通常処理ではない(繰り返し処理など)処理の記述は、インデントをつけたテキスト行

の先頭にコマンドをつけることで表現する。振り分け処理の分岐先など特殊な場合も、インデントとコマンドによるテキストで表現でき、制御線は清書時に自動生成される。

3.2. 主な特徴

HCP Viewer の主な特徴を述べる。

i) 開発言語と動作プラットフォーム

開発言語は Java(1.3)である。JDK1.3 以上の VM が動く環境での動作が期待できる。

ii) コマンド文字変更機能

デフォルトで設定されているコマンドの文字を変更することができる。この機能を利用して、例えば、「Yrepeat」コマンドを「Yr」としたショートカットを追加できる。

iii) データ記述機能

処理記述の行後に「Yin」, 「Yout」コマンドを追加記述することでデータを記述できる。入出力線も自動生成される²。

iv) モジュール分割機能

「Ymodule」コマンドによってモジュールを分割することができる。ひとつのソースファイルにモジュールをいくつも書くことができるので、モジュール分割、統合の試行錯誤的作業が容易である。

v) 表示レベル指定機能

清書閲覧時に、表示レベルを指定して表示できる。レビューの際に便利である。さらに、レビュー時に本処理だけの構造を明示したいことが多いことから、エラーチェック処理の非表示機能を用意している。

vi) SVG 出力機能

SVG(Scalable Vector Graphics)[12]形式で保存できる。SVG 形式からさらに既存のツールを使ってさまざまな画像フォーマットへの変換が可能である。なお、本稿に掲載されている HCP チャートも HCP Viewer を利用して作成したものである。

4. ツールの評価

性能評価実験と運用経験を通して、HCP Viewer の評価を行った。その結果と考察を述べる。

¹ 山本[10]で、制御線を付けたまま接続方向、階層方向へ編集する方法の解説があるが、その概念は直感的でなく、習得するのは難しい

² 但し、データを意図した位置に配置するには、別途配置コマンドを挿入する必要がある。図 5 では配置コマンドを省略している

4.1. 性能評価実験

HCP Viewer と WYSIWYG 方式のエディタ (HCP CHART EDITOR[6])において、HCP チャートの編集時間を測定し比較した。被験者は HCP チャートの作成経験はあるが、計算機支援ツールの経験はない学生 8 名である。

4.1.1. 実験結果

基本清書性能、データ編集性能、トップダウン、ボトムアップ編集性能の各項目について測定を行った。結果を表 1 に示す。実験の詳細は以下のとおりである。

i) ツール学習と基本清書性能

双方の環境を使ったことのない状態から、マニュアルとお手本を片手に住所録管理システム (図 6) を清書してもらい、作業完了までの時間を測定した。それゆえ、この結果には、ツールの使い方を学習するための時間も含まれている。

ii) データ編集性能

データの編集性能を比較するため、既に完成しているチャートに、データを 5 つ追加し、入出力線を 8 本追加する作業の時間を測定した。

iii) トップダウン編集性能

上位 2 レベルが完成しているチャートに、下位に 5 つの処理と、1 つの振り分け処理を追加する作業の時間を測定した。

iv) ボトムアップ編集性能

下位レベルが完成しているチャートに、上位レベルの処理を 5 つ追加する作業の時間を測定した。

4.1.2. 考察

i) WYSIWYG エディタの編集性能について

WYSIWYG 方式のエディタとして PAN/HCP を採用したかったが、動作環境が準備できなかったため、HCP CHART EDITOR を利用した。PAN/HCP のマニュアルより、その他の記述性能はほぼ同様であることが確認できた。但し、

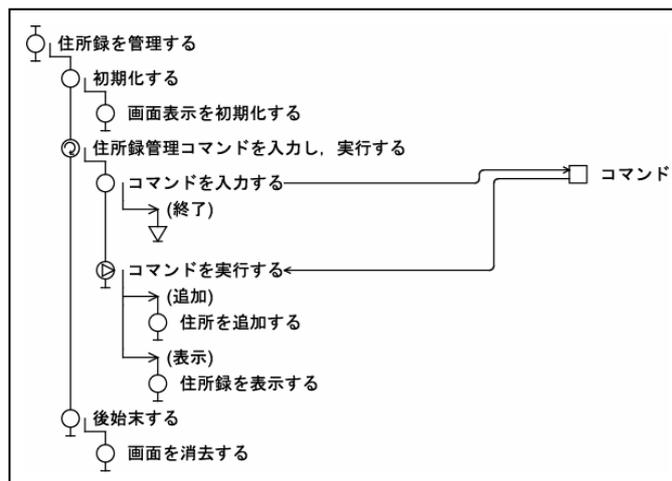


図 6 住所録管理システム

HCP CHART EDITOR はボトムアップの記述をサポートしていないので、ボトムアップ記述の性能比較に関しては、参考程度と考える必要がある。

ii) 作業プロセスの違いについて

WYSIWYG エディタでは、見た目を確認しながら作業が進められるのに対し、HCP Viewer では、一旦清書して、意図どおりの出力になるかを確認する作業が必要となる。しかし、その作業は負担にはならないという回答を得た。逆に、WYSIWYG エディタでは、微妙な位置の調節に集中力が奪われるので、時間がかかるだけでなく、作業後の疲労度も高いことが分かった。

iii) 編集 (修正) 性能について

トップダウン、ボトムアップの編集修正性能に関しては、HCP Viewer により、テキストエディタで編集するほうが圧倒的に早く、疲労度も低いことが確認された。特に WYSIWYG エディタでは、全体的な配置の編集や、それに伴う編集ミスが多数観察された。

iv) データの記述について

データの記述では、WYSIWYG エディタの方が若干書きやすいという意見があった。データの場合は、データの位置と、データと処理

	HCP Viewer		WYSIWYG エディタ	
	平均	範囲	平均	範囲
i) ツール学習, 清書	5 分 0 秒	2 分 27 秒 ~ 8 分 32 秒	14 分 54 秒	10 分 51 秒 ~ 17 分 6 秒
ii) データ記述	2 分 27 秒	1 分 48 秒 ~ 3 分 0 秒	3 分 46 秒	1 分 31 秒 ~ 6 分 54 秒
iii) トップダウン記述	1 分 5 秒	34 秒 ~ 1 分 38 秒	2 分 37 秒	1 分 53 秒 ~ 4 分 24 秒
iv) ボトムアップ記述	51 秒	32 秒 ~ 1 分 10 秒	4 分 50 秒	2 分 2 秒 ~ 14 分 22 秒

表 1 性能評価実験の結果

の対応関係の把握が重要な意味をもつためである。特にデータが多い場合は自動配置に頼ると入出力線が交錯してしまうため、うまく配置しようとするだけでそれだけ多くの時間が費やされてしまう。

速度測定の結果は HCP Viewer も、WYSIWYG エディタとほぼ同等の性能が得られることを示している。これは、HCP Viewer によるデータの自動配置で満足する実験例だったという要因も考慮する必要があるが、5 つ程度のデータ数なら実用上の問題がないことを示している。HCP チャート作成時間の大半を占めるのが処理記述であることを考慮すれば、データ記述はこの性能で十分である。

4.2. 運用経験

4.2.1. 教育現場での運用

HCP Viewer を筆者が担当している設計教育の現場で運用した。主な対象は企業の新入社員研修や大学講義などであり、これまでに使用を経験した初学者は 150 名程になる。

結果、実際の教育現場で、HCP チャートを手書きで作成したことのない初学者さえも HCP Viewer を使って、ほとんど解説なしにチャートを作成できるようになることが分かった。使用レポートも、

- ・ 簡単に、きれいに清書してくれる
- ・ HCP チャートの記法がよくわからなくても、ツールを通して学習することができる

という好意的なものが多い。トラブル報告は、主に出力や印刷関係のものであり、清書コマンドに関する問い合わせはほとんどない。

4.2.2. 実務者の利用

1 年程前からこのツールを研究室 HP にて公開している [13]。筆者に送られてくる使用レポートにより、実務者からの評価も高いという感触を得ている。

5. まとめと今後の課題

本稿では、HCP チャートによる設計教育の支援ツール「HCP Viewer」について述べた。評価実験と運用経験を通して、処理構造の試行錯誤的な編集がしやすく、初学者にも使いやすい環境であることが確認された。データの記述に関しては、既存のエディタと同等の編集性能がみられた。最後に、初学者だけではなく、実務者でも活用できることを確認した。

データの記述性能の向上が今後の課題であるが、これに関しては配置が重要になるため、XYSIWYG 方式の併用も検討されるべきだと考えている。

設計レビューのプロセスを支援するためのネットワーク共有システムとの連携や、ソースコードのコメントからリバースエンジニアリングにより HCP チャートを生成する機能の追加が発展課題である。これらの機能は、教育支援ツールとしてだけでなく、実用ツールとしても重要である。

参考文献

- [1] 花田收悦, 佐藤匡正, 松本匡通, 長野宏宣. コンパクト・チャートを用いたプログラム設計法. 情報処理学会論文誌, Vol122, No. 1, pp. 44-50, 1981.
- [2] 花田收悦. プログラム設計図法. 企画センター, 1983.
- [3] 松澤芳昭, 杉浦学, 大岩元. プログラム設計教育における HCP チャートのレビュー手法. 情報処理学会第 66 回全国大会(1C-2), pp. (4-343)-(4-344), 2004.
- [4] 松澤芳昭, 杉浦学, 大岩元. 学習者同士の相互レビューを通じたプログラム設計教育. 情報教育シンポジウム(sss2004), pp. 189-193, 2004.
- [5] 塩見彰睦, 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, Vol133, No. 2, pp. 183-194, 1992.
- [6]<http://www.vector.co.jp/soft/dl/win31/prog/se007390.html>
- [7]<http://www.dcinc.co.jp/Service/Products/Headway/>
- [8] 佐藤匡正, 浅見秀雄. 処理の論理構造と階層化チャート技法 HCP. 情報処理学会第 23 回全国大会(5J-8), pp. 407-408, 1982.
- [9] 奥出直人, コンピュータが物書きに出会うとき. 1985.
- [10] 山本貴博, 大岩元. ソフトウェア開発支援システム. 情報処理学会研究会報告(ソフトウェア工学-40-5), pp. 25-30, 1985.
- [11] 中神明, 塩見彰睦, 竹田尚彦, 河合和久, 大岩元. HCP チャートエディタの実現と評価実験. 情報処理学会研究会報告(ヒューマンインタフェース-47-20), pp. 149-156, 1993.
- [12]<http://www.w3.org/Graphics/SVG/>
- [13]<http://www.crew.sfc.keio.ac.jp/projects/2004hcpviewer/>