

5R-6

COBOL-AP開発の標準化  
-日本語HCPコンパイラ「CODA」の評価結果-

小林 勝哉 長岡 重輝 池ノ本 利治  
NTT 関西社内情報システム開発センタ

1・はじめに

本稿では、日本語HCPチャート<sup>[1]</sup>からCOBOLソースコードを生成するツールであるCODA<sup>[2]</sup> Ver 1の試行導入評価結果を示す。

評価の結果、CODAはコーディング作業の標準化と自動化を実現し、COBOLソースよりも上位の設計情報であるHCPチャート自体でプログラミングできることから、転記ミスと論理エラーの軽減をはかることが出来ることがわかった。また新人への教育に効果があった。

2・CODA開発経緯

平成元年から社内のAP開発環境の標準化と自動化をめざす開発支援ツールとして、まずHCPレベルのプログラミングツールの開発に着手する事となった。標準化の作業として、約半年にわたり現状分析、表記法統一、記述実験までを行った。また自動化のための機能に不足するものは多いが、現在Ver 1を1990年4月より4プロジェクトに提供して試行評価を行ってきた。

今後は、マルチベンダ環境にも対応可能なAP開発支援方法として、構造化分析手法をベースとする一貫したCASE環境の構築が必要である。したがってCODAに関しても構造化分析/設計手法をベースとする自動化方式を提案<sup>[3]</sup>したところである。

しかしながら、我々の開発するAPは、非常に大規模であるが故に、いかに製品管理を行い、いかに品質の向上をはかるかといった点が課題となっている事は言うまでもない。これらの問題の解決のためには、さらにトータルな開発支援リポジトリの基盤整備が待たれている。

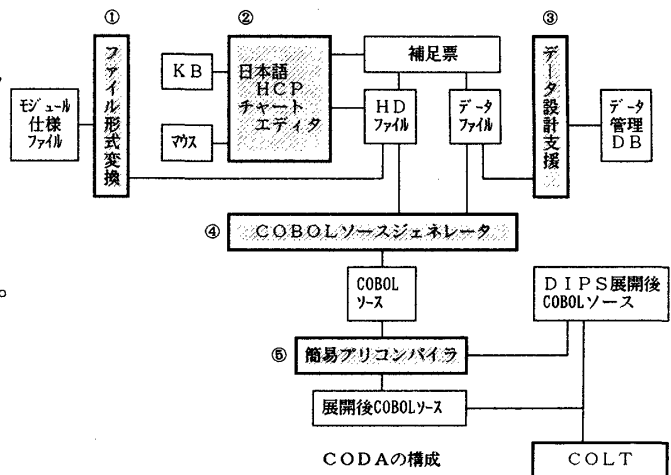
3・処理概要

(1) 機能概要

CODA Ver1は、大別して次の5機能からなります。

- ① テキスト→HCP変換  
上流設計支援ツールで作成した機能仕様書をもとに、HCPチャートに初期変換するファイル形式変換機能
- ② HCPチャートエディタ  
プログラム概略/詳細仕様を記述するHCPエディタ
- ③ データ管理機能  
テーブル/ファイル仕様書を記述し、名票管理を行う
- ④ HCPチャートコンパイラ  
データ部定義ファイルとHCPチャートを入力として、構造化COBOLのソースコードを生成する
- ⑤ ローカル・プリコンパイラ  
1度ホストマシンでプリコンパイルした展開後ソースリストをもとにPCで簡易な手法でマクロ展開を行う

Standardization of COBOL Application Development  
Method - An evaluation of the HCP chart compiler  
: CODA -  
K. KOBAYASHI, S. NAGAOKA, T. IKENOMOTO  
Kansai Internal Information System Development  
Center, Nippon Telegraph And Telephone Corporation



(2) HCPコンパイル性能

PCは、NEC製PC9800シリーズで、PC9801VX2はCPUがi80286、クロックが10MHz、PC9801RA2はCPUがi80386DX、クロックが16MHzである。COBOL命令1Ksあたり(コメントを含むソース行で2K行程度)のコンパイル速度のめやすは以下の通り。いずれもHDD装置は磁気研究所製ML-40。

PCs	HCPコンパイル時間	漢字→か変換時間
PC9801RA2	1 1 2 秒	1 9 6 秒
PC9801VX2	1 5 0 秒	3 8 3 秒

(3) 日本語名票の特性

日本語文の特性としては、多義や省略などにより曖昧になりやすいことがあげられ、CODA Ver1では、限定語彙の分かち書きされた単文で日本語を記述する事とした。

また日本語名票の特性としては、①略記困難、②全角コードによる長大化、③同句異義名票が多く英字名票との対応がアンユニックとなる、などがあげられる。

略記困難な理由は、無理な短縮ではかえって意味不明瞭となるうえ、特徴抽出パラメータが乏しい、記号化しても意味を思考させる、漢字自体が表意である、などがあげられる。

そこで、命名基準が必要となる。命名基準の規定に当たっては、単語の標準化、同一オブジェクト複数表記の排除、意味不明/曖昧な略記の排除、および修飾語の連結記号の標準化を実現できるように留意すべきである。ただし略記法自体はその種類が限定されているため、特別な規約は不要といえる。

比率(%)	日本語名票	英字名票	0	10	20
11.5	予備	FILLER	*****		
5.2	リターン情報	WS4Z3202	*****		
5.0	リターンコード	WS3ZD301	*****		
2.5	ファイル名	WS3ZE114	***		
2.2	詳細情報	WS3ZE403	**		
2.1	内部組織コード	WS3ZC511	**		重複上位5種で25%
1.8	統一組織コード	WS3ZF2142	**		
1.5	レコードキー	WS4Z5B11	*		
1.0	業務コード	WS3ZC112	*		

日本語名票の重複例 (CODA V1)

### 3・評価結果

#### (1) 効果

- ① 新人への教育効果
  - ・プログラミングの基礎研修に効果がある
  - ・COBOL言語仕様のすべてを知らなくてもよい
  - ・生成ソースを勉強すれば構造化COBOLのコーディング技法が自然に身に付く
- ② 若干の品質改善
  - ・名票転記ミスなどのケアレスミスの排除
  - ・単純な論理バグの排除 (IFの入れ間違いなど)
  - ・コーディングの転記作業が省ける
- ③ ドキュメントの標準化
  - ・完全に日本語名標で記述するため、前工程から名標の標準化などの取組みが必要
  - ・HCPレベルの記述規約による日本語の用語の標準化がなされる
  - ・ドキュメントとしての概略HCPとプログラム仕様としての詳細HCPが連動
- ④ COBOLソースコードの標準化
  - ・完全に構造化されたソースコードを生成
  - ・余分な予約語を含まないシンプルなコード
  - ・DIPS構造化COBOLに対応

#### (2) 課題

- ① HMIが不十分
  - ・エディタの操作機能が不十分
  - ・マウスのサポートが不完全
  - ・DOSのメモリネックのため、日本語FEPやLANに制約を受ける
- ② 分かりづらいHCPチャートをメンテしなければならない
  - ・COBOL命令と1対1対応の記述を要する
  - ・ソースコーディングのかわりに詳細HCPを“コーディング”する必要がある
  - ・修正の都度、生成処理時間がかかる
- ③ 大規模の開発のための管理機能が不十分
  - ・1コンパイル単位が複数ファイルで構成される
  - ・HCPファイルがバイナリファイル
  - ・ホスト上のライブラリ化のためには毎回転送を要す
  - ・大規模プログラム開発に関する量的制約をうける
  - ・部品化機能が不十分
- ④ マシンサイトでCODAが使用できない
  - ・ホスト(DIPS)上では漢字の印刷が困難
  - ・PC上のCODA(HCPエディタ、ソースジェネレータ)とホスト上のCOBOLプリコンパイラが連動していない
  - ・リモートデバッグではファイル転送時間がかかる
- ⑤ 開発方法論が未成熟
  - ・ライフサイクル
  - ・モジュール分割
  - ・再利用方法
  - ・ファイル管理
  - ・試験、など

#### (3) 生産性向上指標

自動化への期待は大きいですが、方法論自体はまだ未成熟である。

標準化や、HCP、構造化COBOLなどの基本知識を有しないプログラマに対しては、その教育用システムとしての利用は可能であるが、有技術者に対しては特に直接プラスになる要因はない。

効果を見いだすとすれば、それは標準化にある。CODAにかけるためには、その前工程からの日本語やドキュメントレベルの標準化の取組みが必要となる。したがって、プロジェクト全体として取り組んだ場合に、システム開発工程全体からすれば、後々のメンテナンス時にその効果を期待する事はできる。

また、完全にHCPをソースコードとしてメンテナンスしたいとの希望は多いが、CODA Ver 1ではHCPをソースコード化するために必要な機能が十分ではない。

#### 4・次期CODAの概要

CODA Ver 1の評価結果をもとに検討を進めている次期CODAの概要は次のとおりです。

次期CODAは、構造化分析/設計支援システムのサブシステムとして実現されます。

階層化された日本語文章を入力として、マクロ定義にしたがって、さらに詳細な日本語文章か、COBOLソースに展開するものです。

ドキュメントとプログラム論理との同一チャート上での表記には読解性の点で問題が多く、また、今後のマルチベンダ対応を考慮して、「2階層」仕様記述を提案します。

「2階層」仕様記述とは、概略仕様記述でマシンインディペンデントな仕様を、詳細仕様記述でマシンディペンデントな仕様を記述する事で、ドキュメントとしての仕様書と、マルチベンダ対応可能なプログラム論理仕様とをリンクされた階層化テキストとして管理可能とするものです。

動作環境は、WS上を予定しています。

#### 5・おわりに

貴重な多くのメンバーの協力を得て実施された評価作業により、これまで未知の自動生成系支援システムに対する要求仕様が確立した事は大きな意義があり、関係各位のこれまでの献身的なご協力に対し、深く感謝します。

この評価結果をもとに次期CODAの開発検討を進め、シームレスCASE実現に努力を続けていく予定です。

#### 参考文献

- [1]花田, 「プログラム設計図法」, 企画センタ, 1983
- [2]小林、笠松、塚田、林口, 「COBOL-AP開発の標準化 - HCPチャートベースのプログラミングシステム CODA-」, 第40回情処全大 7S-5 pp1154-5, 1990
- [3]小林、山竹、林口, 「COBOL-AP開発の上流支援 - 構造化分析手法と日本語HCPによる「CODA」の連動に関する一考察-」, 第41回情処全大 1G-9 pp5\_147-8, 1990
- [4]米川、木原、林, 「パソコンにおけるYACIIプログラミングシステムの評価」, 第35回情処全大 5Z-5 pp 1185-6, 1988