

COBOL-AP開発の上流支援

-構造化分析手法と日本語HCPコンパイラ「CODA」の連動に関する一考察-

1G-9

小林 勝哉 山竹 剛 林口 一雄  
NTT 関西社内情報システム開発センター

1.はじめに

本稿では、COBOL-AP開発の上流支援方式に関する一考察をのべる。

上流工程の設計手法として構造化分析/設計手法(以下SA/SDと略記)が定着しつつあるが、我々はCOBOL-AP開発においてもこの手法を本格導入する事を前提に、パイロットAP開発においてその有用性の確認を行った。

一方、COBOL-AP開発の下流支援ツールとしては、日本語HCPチャート<sup>[1]</sup>からソースコードを生成するHCPコンパイラ「CODA」<sup>[2]</sup>を社内で開発し、現在、導入評価中である。これにより、COBOLソースよりも上位の設計情報である、HCPチャートそれ自体をソースとして扱え、論理エラーの減少、保守性の向上、標準化が期待される。

しかし現実には、COBOLソースコードに展開されるCOPY句などのマクロライブラリが既に大量に存在するため、ソースレベルの修正作業からの脱却を困難にする要因となっている。従って、本格的なHCPソースコード化の実現のためには、チャートレベル以上の上位の設計情報にマクロライブラリを移行する事が求められている。

今回、上流/下流それぞれの支援ツールが本格導入されるにあたり、大規模COBOL-AP開発に対して上流から下流まで一貫した分散開発支援方式として、構造化分析の設計情報から、直接日本語HCPチャート、さらにはCOBOLソースコードの生成を可能とする方式について検討したので報告する。

2. SA/SD-CODA連動による上流支援方式

2.1 良い設計書とはなにか

CODAのアプローチは、これまでよりも一步上流の詳細設計情報であるHCPチャートをソースコード化し、標準化することである。しかし、詳細な設計書だけでは「木を見て森を見ず」の結果になりがちである。そこで、計算機支援可能な良い設計書とは何かを検討する。

詳細設計情報のみの記述支援では、リーダビリティ、メンテナビリティ等の点で次のような問題点がある。

- ・全体が見渡せず、本質が見えない
- ・概略仕様と詳細仕様では論理が異なる場合が多い  
(例)サーチ処理は、概略はループ、詳細は振り分け
- ・論理の詳細化が先行するとインタフェース設計を軽んじる
- ・例外処理、ジャンプなどの記述はなくせない
- ・設計言語/チャート・エディタの操作性が重要
- ・何をソースとし、どこまで情報を引き継ぐか不明確

これらの問題点は、実は論理の本質と詳細とが混在している事に起因している。したがって、概略/詳細設計の2段階の立て分けが重要である。

抽象度のことなる設計の階層的支援系に対する要求条件は次の通りである。

- ・概略/詳細情報の2階層以上の抽象度を扱える事。  
これにより、必要な詳細化レベルに応じた設計書を記述でき、理解しやすくなる。例えば、論理の本質はSA/SD手法のプロセス仕様で、論理の詳細はプロセス仕様にリンクした詳細HCPで記述する等である。
- ・設計情報を自動引継(転記レス)する事。  
転記が必要な間は、スキルにより記述レベルが異なるため、標準化ができない。
- ・概略設計情報でのコンパイルが可能な事。  
上流でのマクロ記述が可能であれば、概略記述でのコンパイルが可能となる。そのためには、プロセス仕様設計言語による部品、またはHCP記述部品のユーザー定義、自動展開機構が必要である。

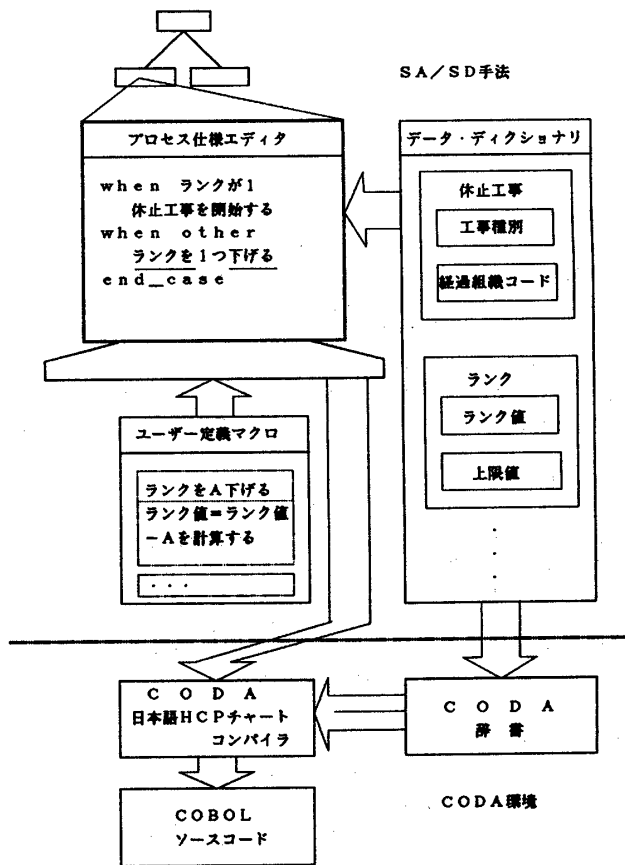


図1 SA/SD-CODA連動概念図

Upper CASE for COBOL Application Development :

- A study on connenction SA/SD Method to CODA :A Japanese HCP Chart Compiler -

K. KOBAYASHI, T. YAMATAKE, K. HAYASHIGUTI

Kansai Internal Information System Development Center, Nippon Telegraph And Telephone Corporation

## 2・2 我々の提案する方法

上流から下流まで一貫した分散CASE環境として、SA/SD手法と日本語HCPコンパイラ「CODA」の連動によるコード生成方法を提案する。これにより、コーディング作業が上位の設計作業へ移行し、ジェネレータ方式にないユーザーカスタマイズが可能となる。

連動方式としては、次の「プロセス仕様ユーザー定義マクロ」を提案する。COBOLのCOPY句などに代表されるマクロ機能を、プロセス仕様書レベルで実現する方式である。

### 「プロセス仕様ユーザー定義マクロ」の概要

#### ① CODA標準構文

(CODA Ver 1でサポート済み)

#### ② データ宣言の引用

(エディタにおけるデータディクショナリの参照、引用)

#### ③ ユーザー定義構文

・日本語キーワード定義構文  
(C言語のプリプロセッサ機能に相当)  
(定義は詳細日本語HCPレベル)

・処理定義構文  
(マクロ機能に相当)  
(記述は詳細日本語HCPレベル)

#### ④ プロセス仕様書部品化機能

(上記データ宣言、ユーザー定義構文の外部定義)

## 2・3 定性的評価

### ① 日本語表記マクロの評価

米川ら<sup>[3]</sup>による日本語表現でのマクロ記述の評価結果は、おおむね良好である。ただし、プロジェクトとしてマクロの一元管理と標準化が重要である。

### ② 構造エディタの評価

篠塚ら<sup>[4]</sup>をはじめとする構造エディタによる設計の評価結果は、設計テキストとプログラムコードが混在した無階層エディタとして、その有用性が報告されている。しかし、無階層では言語依存を脱却できない。上流設計情報を記述するためには、我々の提案するプロセス仕様記述言語の標準化とマクロ機能による抽象化、及び日本語で詳細設計を行う階層化構造エディタが必要となる。

### ③ 日本語名票の一元化の評価

SA/SD手法ではD/Dをもとにデータ設計を行うため、大規模AP開発においては、日本語記述のデータ名の衝突を如何に解決するかが、ポイントである。

既存の大規模オンラインAPの共通データ仕様書に記述された日本語名票の命名特徴分析の結果、次のように分類できた。このバリエーションを克服するためには、日本語名票命名基準の標準化が重要である。

## 3・あとがき

以上、上流から下流まで一貫したAP開発方法として、SA/SD手法の生産物であるプロセス仕様書レベルでのユーザー定義マクロ機能と、D/Dの引用により下流支援ツールの「CODA」に連動する考え方を紹介した。今後は、実際にエディタ上にマクロ機能を実現し、評価実験を行っていく予定である。本方式により、SA/SD手法のメリットを生かしつつ、実際の大規模AP開発に適用しうる分散CASE環境の構築が可能であると考えている。

|                    |          |
|--------------------|----------|
| ・プレフィックス (数字、英字など) |          |
| 1・ランクC (下限値)       | DKQ01LKC |
| 1・設計発生時期           | CJHSKI01 |
| ・サフィックス (数字、英字など)  |          |
| 管理表示・1             | CJBKNR01 |
| 管理表示・2             | CIKRRHY2 |
| ・サフィックス (括弧注記)     |          |
| ランクA (下限値)         | DKQ01LKA |
| 受委託コード (施工)        | CE5JYCDS |
| ・汎用語               |          |
| その他                | CCOKJKB8 |
| 年月日                | CNHNENGP |
| ・類似語               |          |
| 工事種別               | CCOKJKB4 |
| 工事別                | CCOKJKB6 |
| ・各種コード             |          |
| 固定資産組織コード          | CG6KZSCD |
| 計画組織コード            | CKOKESCD |
| ・各種区分、種別           |          |
| 任意区分               | CKONIKUB |
| 工事種別               | CDOKJSHU |
| ・ドットペア             |          |
| 建・損別               | CJ2KSBET |
| 請・直別               | CJ2UTBET |
| ・計算式               |          |
| N+1 (予定)           | DG6JGEN1 |
| 集約番号 N+1           | CJKSSBP1 |
| ・分かち書き             |          |
| 休止工期 開始            | CCOJKYUK |
| 休止工期 終了            | CCOJKYUE |
| ・特殊コード             |          |
| 比率⑦                | DS1SGNH7 |
| 契約額巾 (~迄)          | DJUKYKGG |
| ・あいまい              |          |
| 工事場所又は、区間名         | NJ4KJBME |
| 労務費 (再掲)           | DCOKSRZU |
| ・全角半角混在            |          |
| 稼働実績 (建設)          | DCOKSKAD |
| 稼働実績 (建設)          | DJ2KSKAD |

図2 日本語名票命名特徴分析パターン

## 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]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
- [4]篠塚、下村、磯田, 「拡張可能テンプレートを用いた設計書・コード作成支援エディタ」, 第36回情処全大 1N-2 pp1081-2, 1988