

## 構造化ソフトウェア開発システム

### “(CANDO)”

ファコム・ハイタック株式会社 (ハイタック本部)      大野 治      松本 彦三

#### 1 はじめに

ソフトウェアの信頼性及び生産性を高める方策としては、仕様記述言語に代表される超高級言語を利用して毎回新たに目的ソフトウェアを効率よく建設する方法と、ソフトウェアを部品化しておき、それを再利用する方法とがあり色々と提案されている。部品の再利用についても従来から提案及び実用化されているが、それらの部品は経験的に集められたままの状態で運用されている。ファコム・ハイタック株式会社 (ハイタック本部) では、既存プログラムの集合から類似部品を抽出整理し、その部品を実績データで検証した。その検証された部品群をベースとした構造化ソフトウェア開発システムを開発し、このシステムをCANDO (Computer Aided New facilities for system Development and its fixation to the Organization) と名付け実用化している。

本報では、CANDOシステムの開発背景及びシステム構成について概説するとともに、CANDOシステムの適用結果とその評価についても合わせて報告する。

#### 2 CANDOの開発背景

##### 2.1 CANDO開発の動機

ソフトウェアの重要性が認識され始めたころから、ソフトウェア開発方式の改革、改善の努力がなされてきた。トップダウン設計などに代表される設計技法の開発、あるいは部品化技術などの製造技術の開発も随所で実施されてきた。これらは、ソフトウェア開発工程の一部分に対し効果をもたらしたが、ソフトウェア開発のCAD化というテーマを解決するには至らなかった。この最大の理由は、設計書や部品などを磁気化して蓄積したり、これらをインタラクティブに検索できるような、機械設備が整わなかったことによるものと考えられる。最近のファイルの大容量化、タイムシェアリングシステムの普及は、これら問題点の解決を可能にし、更にソフトウェア開発の全工程に及ぶCAD化の実現も可能にした。

CANDOシステムは、以上の背景のもとに開発したものである。

##### 2.2 CANDOの開発方針

CANDOのねらいは、ソフトウェア開発の全工程に対し、標準化と機械化を徹底し、その開発方式を組織内に定着化することにある。この目標達成のために、以下のような開発方針を設定している。

(1) ソフトウェア開発の全工程を標準化し、標準化を前提にした

各種のソフトウェア開発支援ツール、ソフトウェア部品を用意する。

(2) これらソフトウェア開発支援ツール、ソフトウェア部品は、すべて会話環境下で使用できるようにする。

- (3) ソフトウェア部品群は、過去に蓄積してきたものを整理、統合し、真に必要なものだけを登録する。また、個別プロジェクトの開発環境の変化に即応して、追加、削除、改良などを行なえるような仕掛けを提供する。
- (4) 普及活動には特に留意し、実務と密着した教育活動を積極的に推進する。また、CANDOに対する生産現場からの改善提案を吸い上げられるような組織作りも合わせて推進する。
- (5) 標準化については、日立製作所のアプリケーションシステムの効率的設計技法（H I P A C E : Hitachi Phased Approach for High Productive Computer System's Engineering)を基準とし、ツール群は、それを補完する立場に立って開発する。

### 2.3 CANDOシステム構成

CANDOシステムは、図1に示すようにシステム設計支援からドキュメント出力支援まで、ソフトウェア開発の各工程を対象としている。各工程を支援するサブシステムは、相互に密接に接続され一貫したものとなっている。

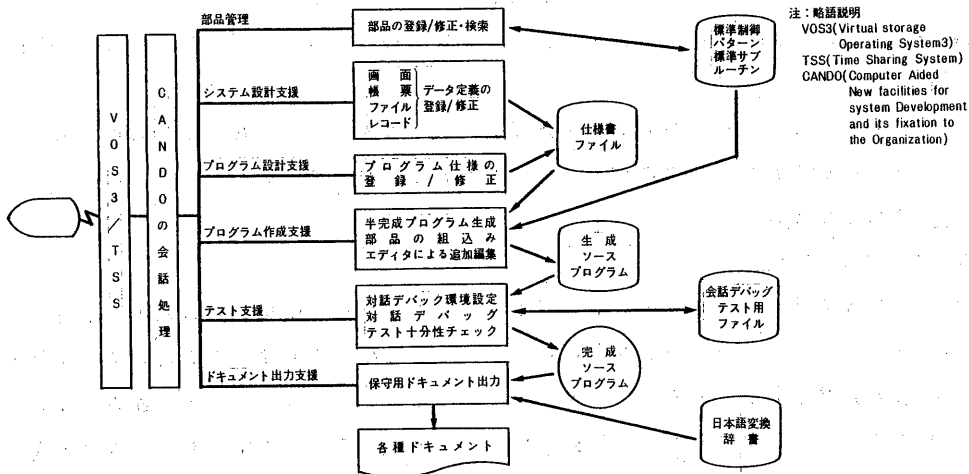


図1 CANDOシステムの構成

### 3 CANDOシステムの機能

CANDOの代表的なソフトウェア開発支援ツールの機能及び特長について次に述べる。

#### 3.1 システム設計支援機能

##### (1) 画面設計支援

画面のイメージを対話形式で端末から入力することにより、業務プログラムで使用する XMAP (Extended Mapping Aids) パラメータを生成し、目的画面レイアウトを作り、これを CRT (Cathode Ray Tube) 画面に表示、又はプリンタに印刷する。この結果により、仕様の確認が可能になる。さらに画面間の遷移情報を対話形式で入力することにより画面遷移の確認も可能になる。

## (2) 帳票設計支援

帳票イメージを対話形式で端末から入力することにより、書式制御を含めた本番印刷イメージの帳票をプリンタに出力する。この結果により、仕様の早期確認が可能になる。

## 3.2 プログラム作成支援機能

### (1) 半完成プログラム生成

プログラムの制御構造を規定する標準制御パターンと入出力部品を合成し、半完成プログラムを生成する。この際、画面設計・帳票設計で定義した内容、及びプログラム設計で定義した仕様書ファイル上にある情報などを半完成プログラム上へ反映する。同時に、半完成プログラム上のコードを、あらかじめ定義したユーザ固有名称に変換する。

### (2) 部品の組み込み

ユーザ固有のプログラム部品、又はCANDOで準備した標準的なプログラム部品を前項(1)で生成した半完成プログラムに自動的に組み込む。このときユーザ固有名称の変換も同時に行なう。

## 3.3 テスト支援機能

### (1) テスト十分性指標(C<sub>1</sub>、S<sub>1</sub>メジャ)の採用

プログラムの実行順序の組合せは天文学的な数となり、そのすべてをテストすることは不可能である。この中から実用上十分なソフトウェア品質を保証し、かつ実施可能なテスト項目を抽出するため、テストカバレジメジャとして、C<sub>1</sub>メジャ、S<sub>1</sub>メジャを採用した。対話テスト環境の中で、C<sub>1</sub>メジャ又はS<sub>1</sub>メジャを測定し、テスト進捗度の把握及びテスト十分性の評価実施を支援する。

### (2) 対話テスト環境の自動設定

単体テスト支援では、ソースプログラム情報を解析し、自動的に対話テスト環境(実行中断点及び外部プログラム、ファイル入出力のシミュレートなど)を設定する。また、組合せテスト支援では、テストデータを実ファイル上へ自動作成する。テストデータの追加、修正を対話形式で可能にするため、テストデータ編集機能もサポートしている。

## 3.4 ドキュメント出力支援機能

ドキュメント出力支援機能を用いると、ソースプログラムから日本語で記述されたドキュメントを出力できる。出力ドキュメントとしては、プログラム関連図、プログラム機能説明図、セクション関連図、NSチャート、PAD図データ定義書、日本語ソースリストなどがある。

## 3.5 プロジェクト管理支援機能

プロジェクトの進捗管理上必要となる評価項目及びその尺度を定め各評価項目に関する実績データを、個々の作業担当者の意識しない状況で自動的に収集する機構を実現している。この機構により収集したデータをもとにプロジェクト全体の進捗状況表及び品質管理手法を活用した各種アラーム情報をプログラム毎に出力している。

- 
- \* 1) C<sub>1</sub>メジャ：単体プログラムテストで、すべての分岐方向のうち少なくとも1回実行されたものの割合。
  - \* 2) S<sub>1</sub>メジャ：組合せプログラムテストで、各プログラムへのすべての呼出し命令(CALL)のうち少なくとも1回実行されたものの割合。

#### 4 設計標準と部品

CANDOでは設計作業の標準化を図るために、ソフトウェア標準設計書及び部品群を準備し活用している。

##### 4.1 CANDOを構成するソフトウェア部品群

###### (1) 標準制御パターン

COBOLプログラムを中心に、バッチ処理形態、オンライン処理形態ごとに標準制御パターンを整備している。(COBOL以外にはPL/IとNHELPを用意している)

###### (2) CANDO標準部品及びユーザ固有部品

CANDO標準部品としては、日付処理、チェック処理、文字列処理など、使用頻度の高いプログラムを部品として整備している。また、ユーザ固有部品についても、各サイドで業務に共通して使用される機能、ロジックなどを部品として登録、管理することができるようにしている。

##### 4.2 構造化設計の徹底

(1) 構造化日本語で記述された標準設計書を準備し、これを活用して個々の設計書を作成

(2) 登録した標準制御パターンは、すべて構造化したプログラム構造

(3) NSチャートの適用及び「GO TO LESSプログラミング」の推進による構造化プログラミングの徹底

(4) 各種静的解析ツールを用いた構造化設計の検証

昭和59年上期末までにCANDOを適用して開発したプログラムの規模は数百万ステップに達した。この適用過程で得たノウハウの蓄積を部品及びツールにフィードバックさせてきた。

#### 5 適用結果

昭和56年下期からCANDOの適用を開始し、次にその適用状況及び結果について述べる。

##### 5.1 CANDO定着化状況

昭和56年下期開始当初全プロジェクトのうち30%のプロジェクトに適用したが、昭和59年上期では80%と3倍近い定着化率を示している。また、社内外のCANDO既教育人数は、昭和59年上期末現在1,000名を越えている。以上から、ここ2年の間に急激な定着化が図られたことがわかる。

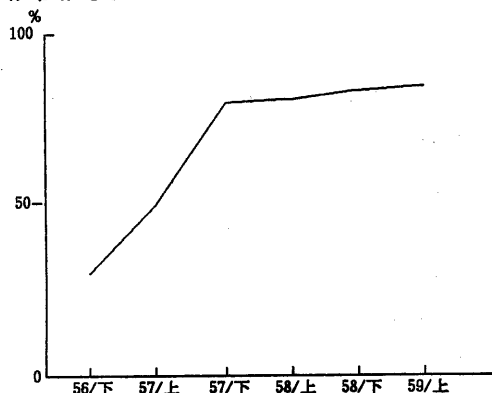


図2 CANDO定着化状況

## 5.2 設計製造の分離

設計及びプログラミング工程の標準化により、工程間の情報伝達が可能になったため、設計部門とプログラミング部門(製造部門)の分離を実現した。これはソフトウェア生産の工業化が可能になったということである。当然のことであるが、組織を分けることによる設計品質及びプログラミング品質に対する相互牽制は、急速なシステム習熟に寄与している。またプロジェクト管理機能を使用することにより

① 管理者が作業の実態を把握できるようになった。

② 作業担当者の異常やトラブルが早期に発見できるようになった。

以上のことから(製造部門作業)の外注化が可能になり、現在この生産形態を推進中である。

## 5.3 信頼性の向上

次に、従来方式とCANDO方式との信頼性比較を示す。CANDOでは、設計品質、プログラム品質の向上を図るため、設計時でのデザインレビューの徹底、プログラム作成時でのテスト十分性指標の採用を実施している。特に構造化の徹底、 $C_1=100\%$ を実現している。図5は、単体テスト時に発生するバグについて、CANDO方式と従来方式とで比較したものである。CANDO方式のほうがバグ密度が小さく、またCANDOの機能拡張及び使用経験が増すことにより、さらにバグ密度が小さく、しかも分散が小さいことがわかる。また、単体テスト時でテスト十分性指標を用いたテスト( $C_1$ メジャ=100%の達成)を実施しているため、後工程(組合せテスト、運用テストなど)へ積み残すバグが減少している。

従来方式のバグ密度平均を100とした単体テストにおけるバグ密度分布

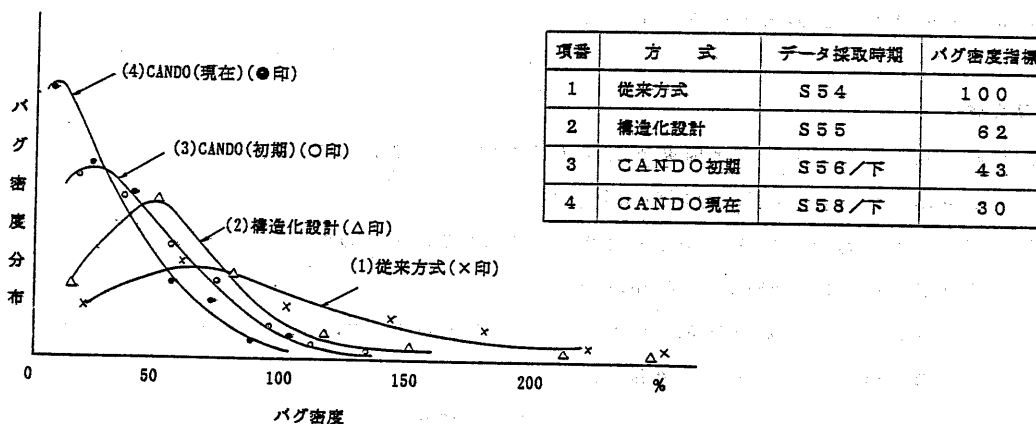


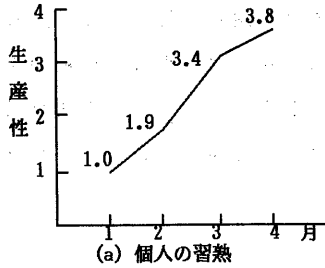
図3 単体テスト時のバグ密度分布

## 5.4 生産性の向上

CANDOではプログラム作成を支援する各種ツールや標準設計書を使用することにより、生産性の向上を図っている。図4はCANDO適用時の生産性向上に関する習熟過程及びCANDO使用時、未使用時の工程別工数分布を示している。同図(a)は個人の習熟であり、約3か月で習熟傾向を示している。しかも、開始当初の約3倍の生産性を示している。

また、同図(b)から主に組合せテスト、運用テストなど後工程の作業での工数が削減されて

いることがわかる。



		工数		
CANDO 未使用	システム設計	プログラム設計～ 単体テスト	40%	工数 (1.0)
	20%	テスト	CD～SD 40%	
CANDO 使用				工数 (0.45)
		38%	36%	
	17	16	12	

(b) システム開発工程の作業比

図4. CANDOの適用による生産性の向上

## 6 結 言

CANDOシステムは、ソフトウェア開発の全工程を対象としていることによる総合性と、これを構成する各サブシステムが相互に密接な関連をもつことによる一貫性に特徴があり、ソフトウェア開発の近代工業化の観点から、大きな効果を発揮するものと期待している。今後、更に、ソフトウェア開発の機械化率を高める手段を求め、CANDOシステムの機能拡張及び操作性向上を目指した機能改善の推進に努力する。

## 参考文献

- 1) 片岡, ほか: ソフトウェア開発支援システム(CASDシステム),  
日立評論, 62, 12, 879~882(昭和55-12)
- 2) 小野, ほか: 構造化ソフトウェア開発システム“CANDO”,  
日立評論, 66, 3, 195~198(昭和59-3)
- 3) 葉木, ほか: システム開発支援ソフトウェア“EAGLE”,  
日立評論, 66, 3, 189~194(昭和59-3)
- 4) G.J.Myers著(長尾訳): ソフトウェア・テストの技法,  
近代科学社(昭和57-10)
- 5) 大野, ほか: 構造化ソフトウェア開発システム“CANDO”によるプログラム品質の  
向上,  
日科技連 第4回 ソフトウェア 生産における品質管理シンポジウム61~66(昭和59-9)