

構造化ソフトウェア開発システム“CANDO”

Structured Software Development System “CANDO”

増加の一途をたどるソフトウェア開発で、生産性、保守性及び信頼性の向上を確保するためには、従来のような前近代的な生産方式では対処が難しく、ソフトウェア生産方式の抜本的革新が必要になってきた。ファコム・ハイタック株式会社(ハイタック本部)では、ソフトウェア開発のCAD化を目的としたCANDOシステムを開発し、実用化した。

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、その開発方法を組織内に定着化することである。

CANDOを個々のプロジェクトに適用した結果、従来方式に比べ信頼性の高いソフトウェアを効率良く開発することが可能となった。

小野 功* Isao Ono

大野 治* Osamu Ôno

天明宏行* Hiroyuki Temmei

1 緒 言

ハードウェア技術の飛躍的進歩により、コンピュータシステムの原価構成は、ソフトウェア中心へと急速に移りつつある。また、ハードウェアの技術革新に支えられて、コンピュータの利用形態も年々高度化、多様化の一途をたどっている。このような環境変化に対し、従来のような前近代的なソフトウェア生産方式では対応することが難しく、ソフトウェア生産方式の抜本的革新が必要になってきた。言葉を換えると、ハードウェア開発で採用しているCAD(Computer Aided Design)化の考え方を、ソフトウェア開発にも適用すべき時期にきたということである。

ファコム・ハイタック株式会社(ハイタック本部)では、市場の要請にこたえるために、ソフトウェア開発のCAD化を目的とした、CANDOシステム(Computer Aided New facilities for system Development and its fixation to the Organization:構造化ソフトウェア開発システム)を開発し、実用化した。

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システムを構成する代表的なソフトウェア開発支援ツール、及びソフトウェア部品を紹介する。また、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の概要

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¹⁾、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、その開発方法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た。

- (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) ソフトウェア開発支援ツール、ソフトウェア部品群、ドキュメント体系は、いずれも構造化に徹したものにす。
- (5) 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。
- (6) 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(HIPACE: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のシステム構成

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²⁾。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ており、各サブシステムを構成するソフトウェア開発支援ツールも密接に連結され、一貫したものとなっている。

* ファコム・ハイタック株式会社

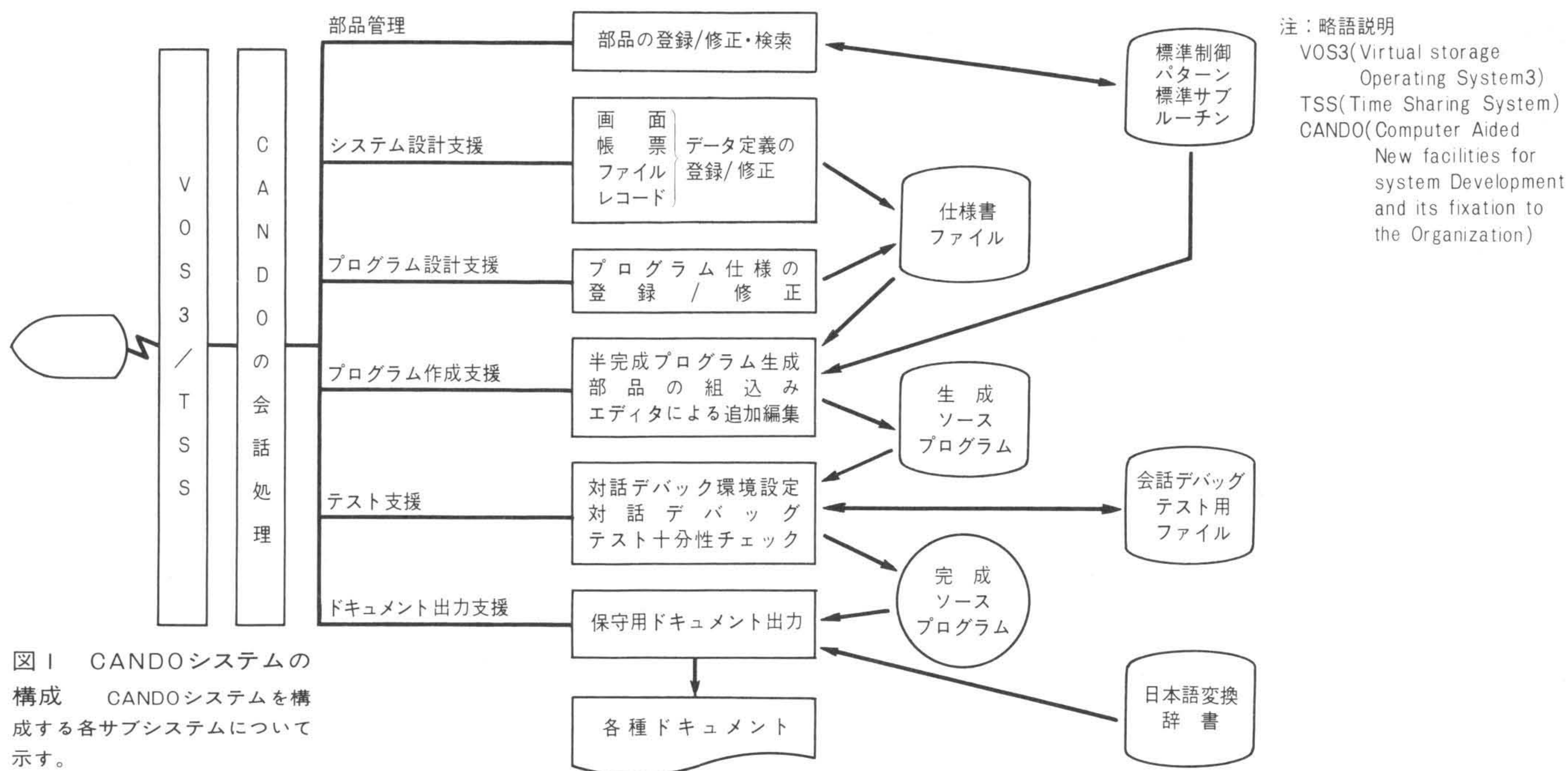


図1 CANDOシステムの構成 CANDOシステムを構成する各サブシステムについて示す。

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徴について次に述べる。

3.1 システム設計支援機能

システム設計支援機能として、画面設計支援と帳票設計支援の二つの機能がある。それぞれは、コンピュータ処理のアウトプットである画面及び帳票についてレイアウトの作成を支援するものである。

(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。

(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仕様書ファイル上に定義して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ー固有名称に変換する。

(2) 部品の組込み

ユーザー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。このときユーザー固有名称の変換も同時に行なう。図2に部品の組込み方式を示す。

3.3 テスト支援機能

(1) テスト十分性指標(C₁, S₁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

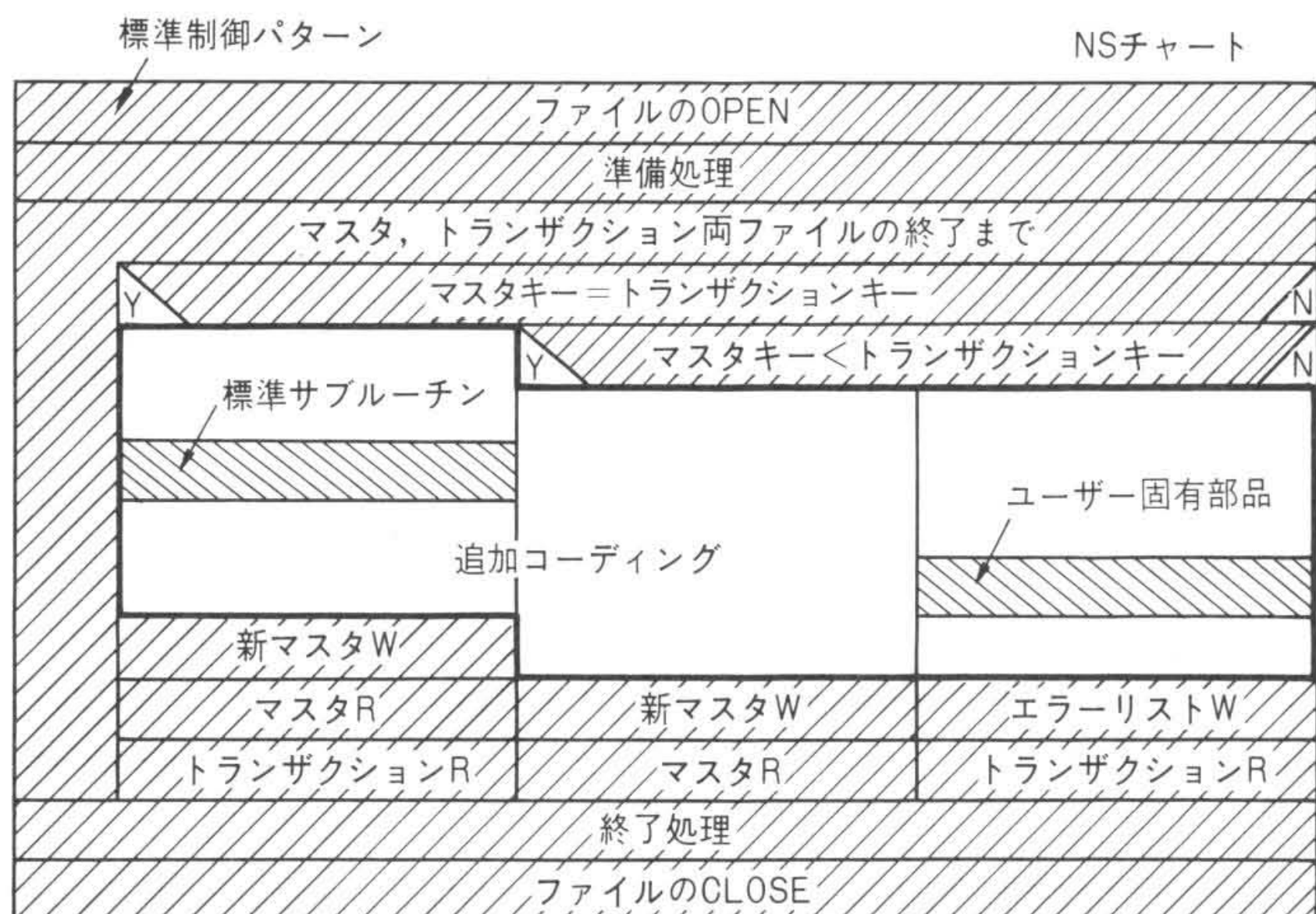


図2 部品の組込み ユーザー固有部品、標準部品の半完成部品への組込み方式について示す。

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、テストカバレッジメジャ³⁾として、C₁メジャ^{*1)}、S₁メジャ^{*2)}を採用した。図3にC₁メジャについて示す。

対話テスト環境の中で、C₁メジャ又はS₁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。対話テストは日立製作所製プログラムプロダクトCOBDBG(COBOL Debugging Aids)の下で行なう。

(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

※1) C₁メジャ：単体プログラムテストで、すべての分岐方向が少なくとも1回は実行される。

※2) S₁メジャ：組合せプログラムテストで、各プログラムへの呼出し命令(CALL)がすべて実行される。

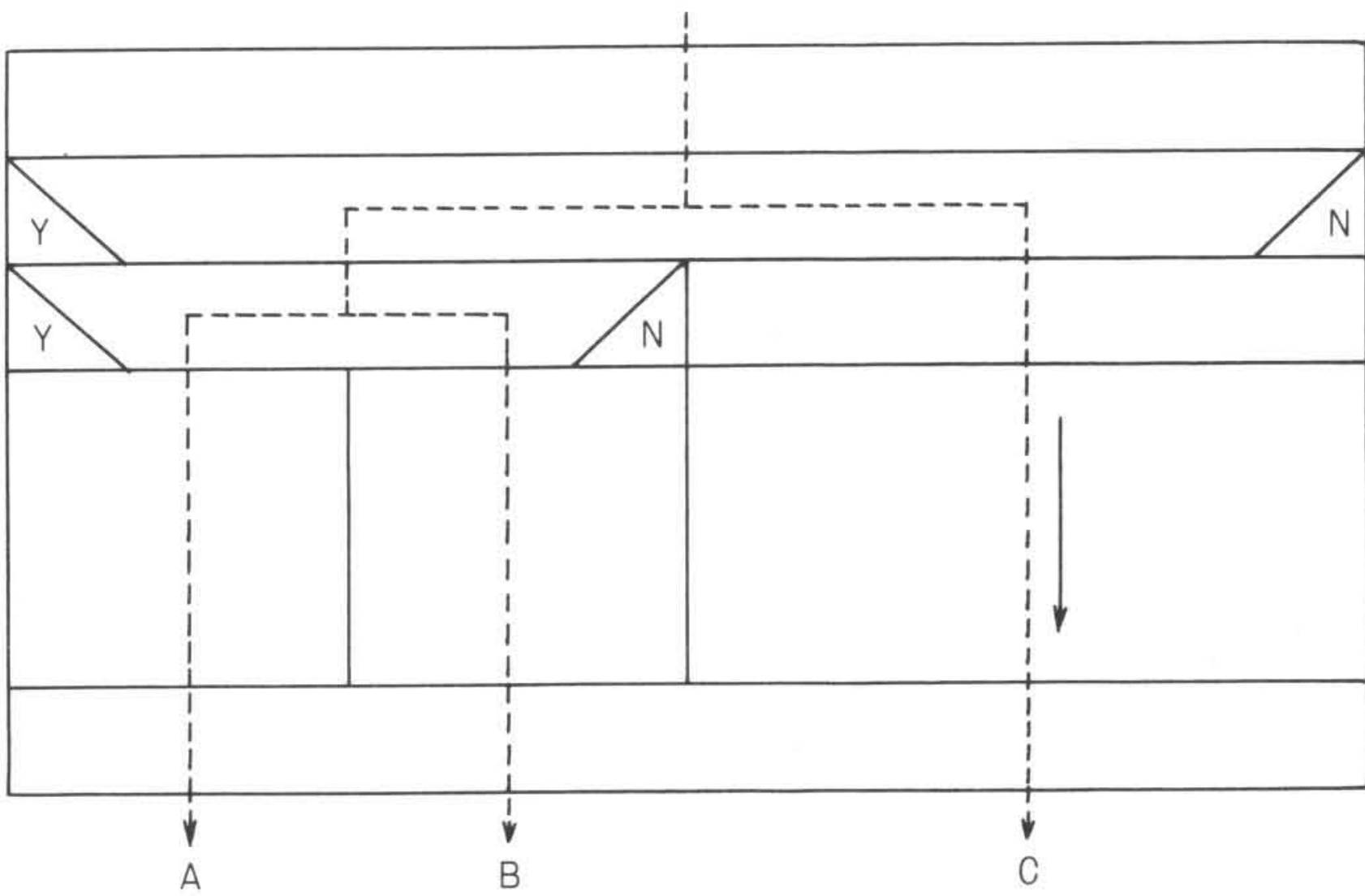


図3 テスト充分性指標C₁メジャ 本図は、プログラムの手続き部の一部をNSチャートで示したものである。IF文などで発生する径路(A, B, C)について実行し、分岐網羅率を測定したものがC₁メジャである。つまり、プログラム論理を実行する上ですべての分岐を網羅したとき、C₁メジャ=100%となる。

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 エディット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、日本語ソースリスト、プログラム機能説明図、NSチャート、データ定義書などがある。サンプルとして図4にNSチャートを示す。

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。標準設計書は、すべて構造化された日本語で記述されている。

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。

(2) CANDO標準部品及びユーザー固有部品

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、ユーザー固有部品についても、各サイト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

CANDOでは、次に述べるような方法により構造化設計の徹底を図っている。

- (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する。
- (2) 登録した標準制御パターンは、すべて構造化されたプログラム構造とした。
- (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

5 適用結果

昭和56年下期からCANDOの適用が開始された。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

図5にCANDOの定着化状況について示す。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58年上期では83%と3倍近い定着化率を示している。また、社内外へのCANDO既教育人数は、昭和58年上期末現在では、800名以上に達している。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分かる。

5.2 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6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箇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。

一方、プロジェクトでも、習熟効果が現われており、初期段階で既に約2倍の向上が実現されている。

また、同図(c)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分かる。

5.3 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

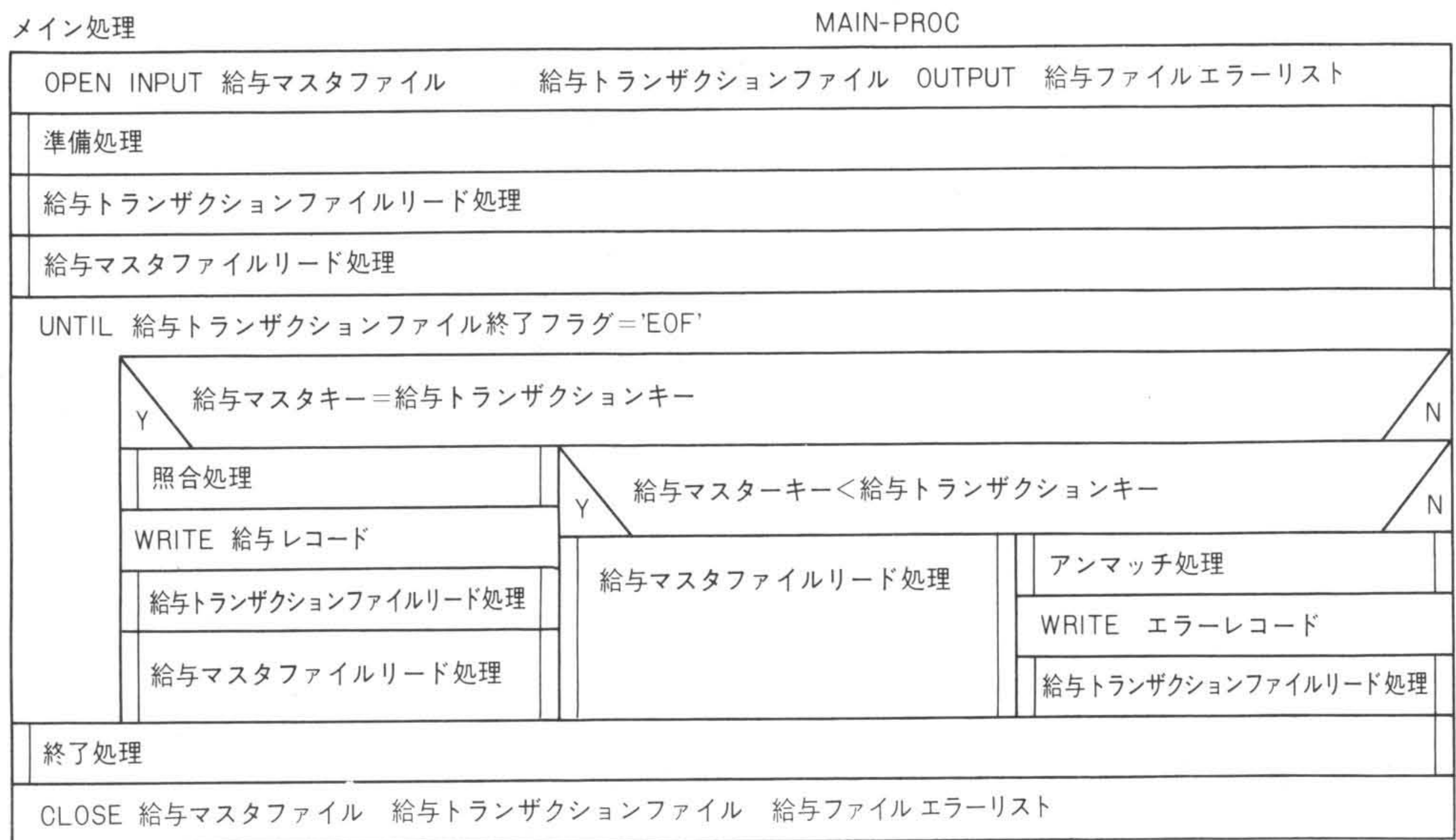
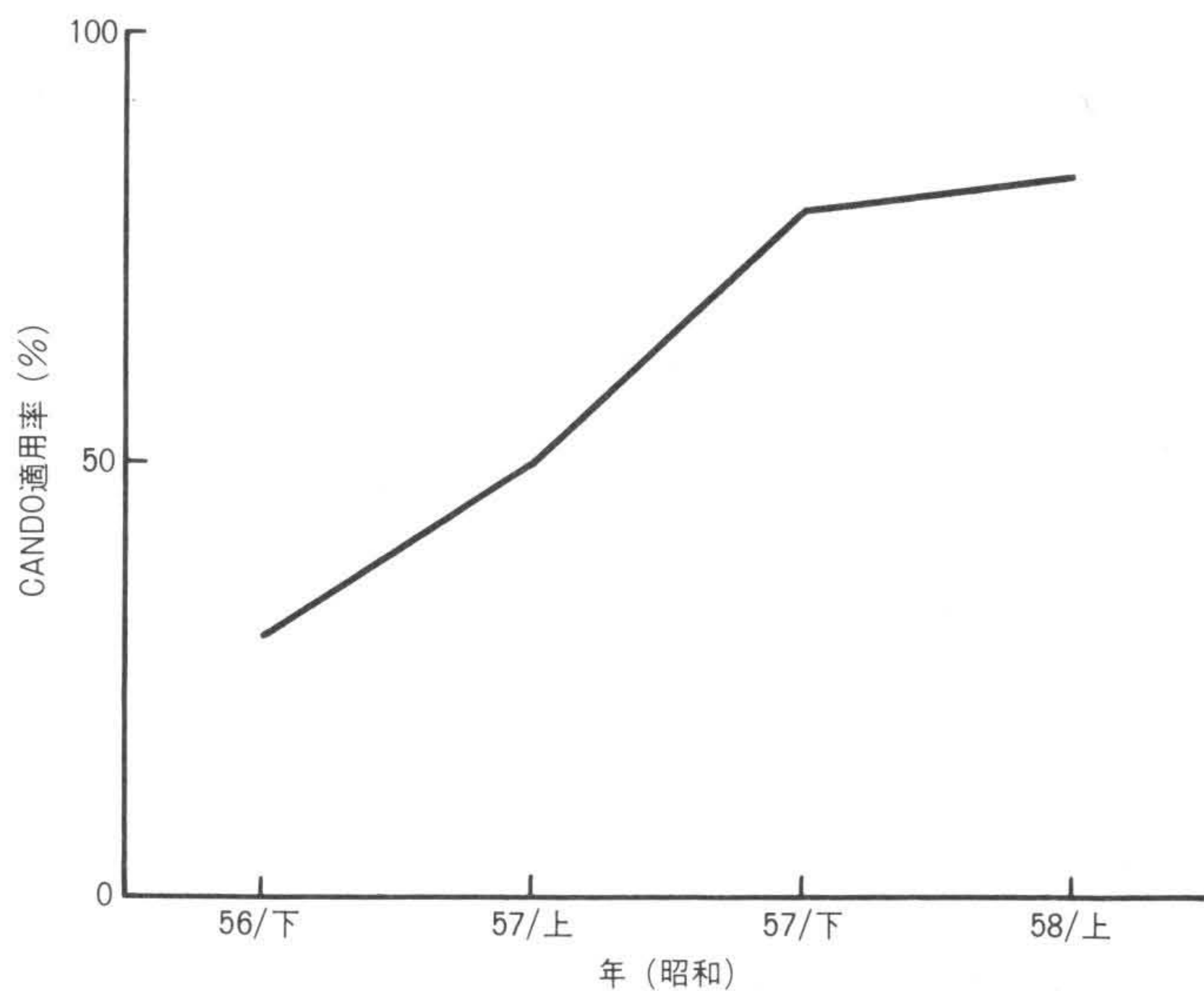
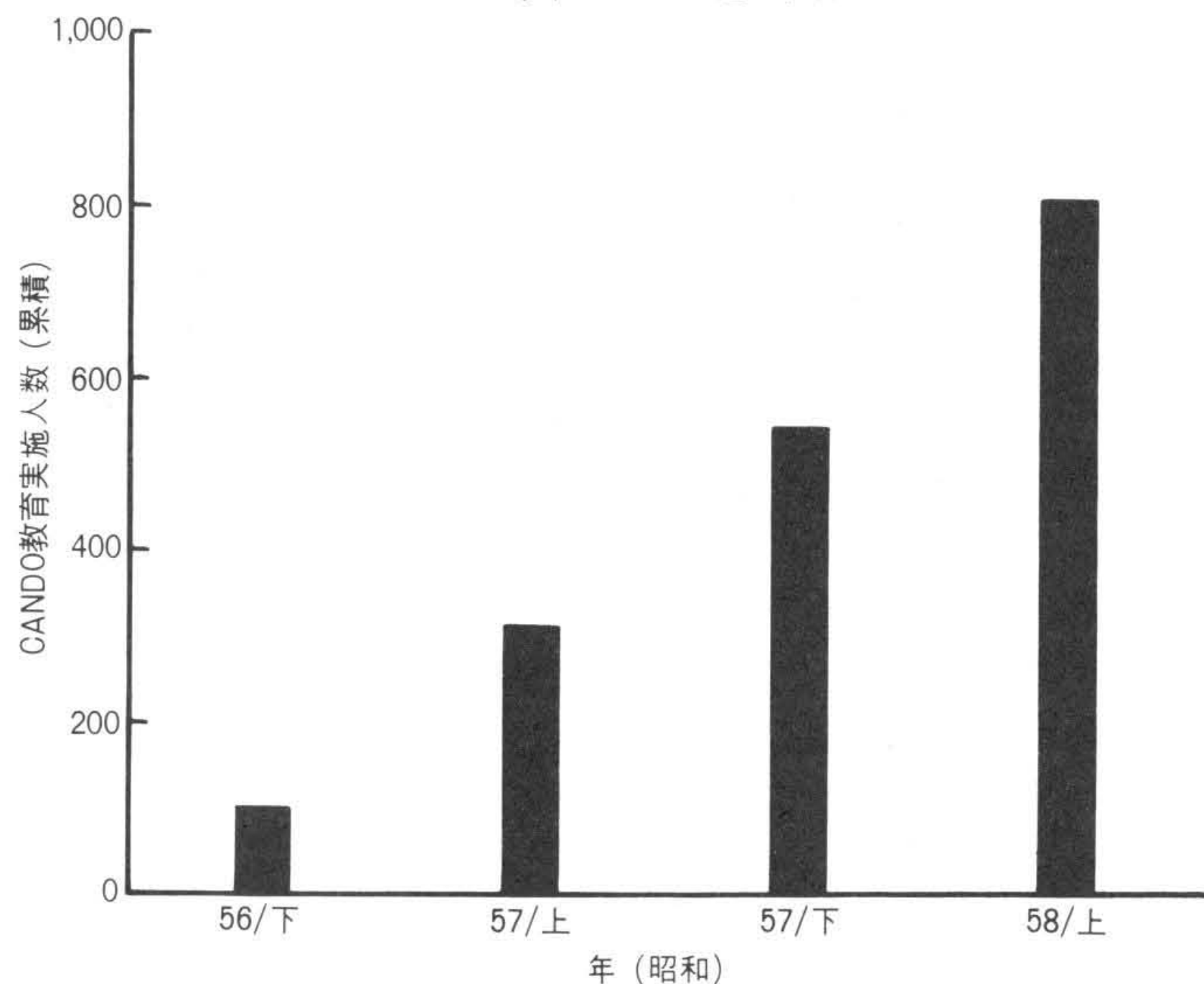


図4 NSチャート ドキュメント出力支援機能を用いて出力したNSチャートを示す。

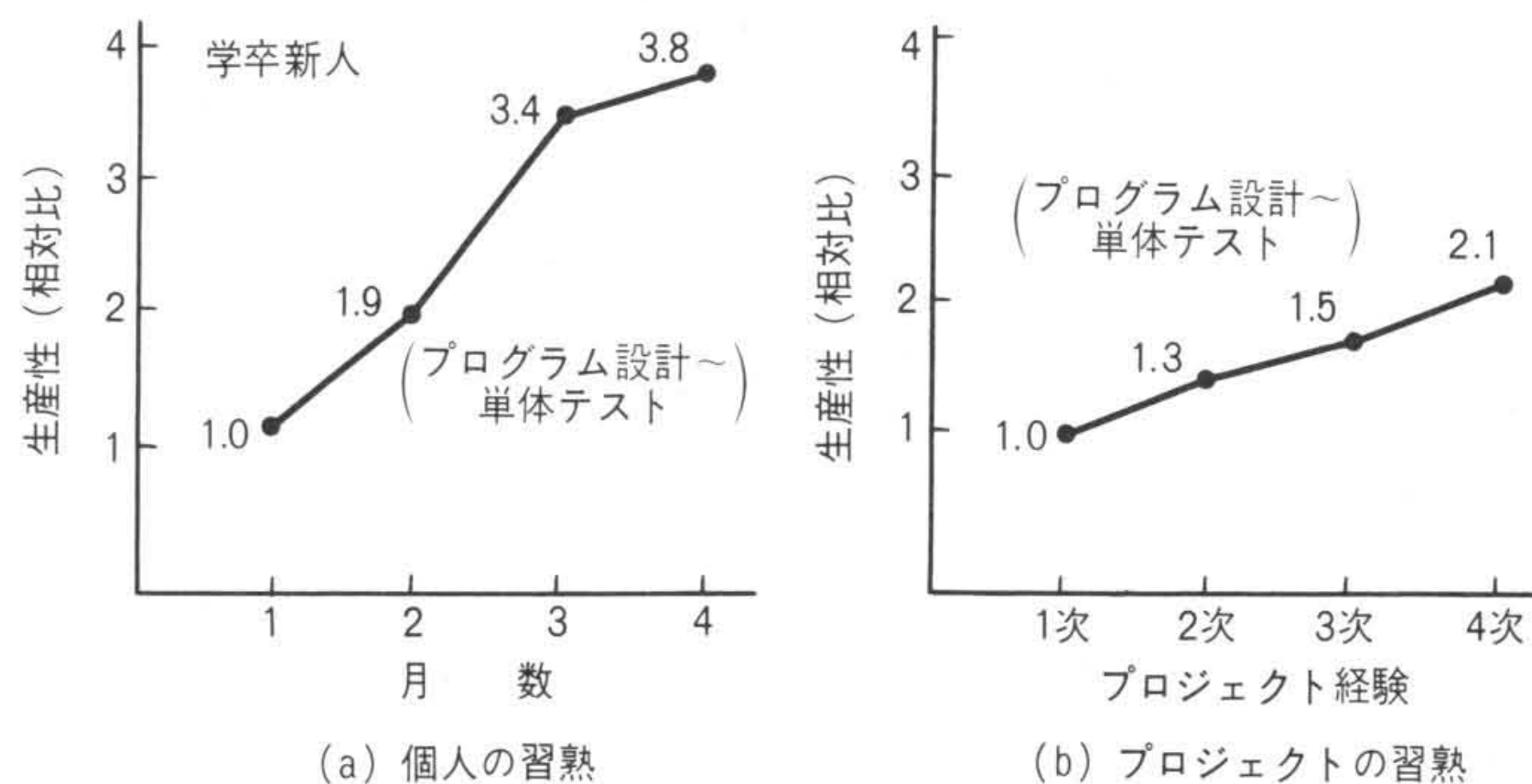


(a) CANDOの適用状況



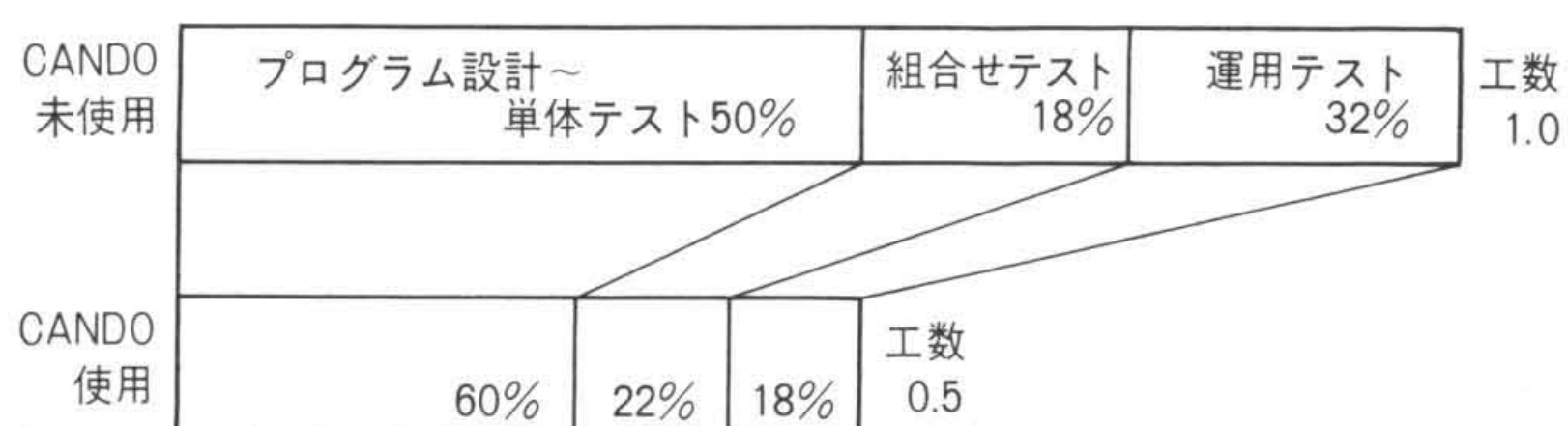
(b) CANDOの教育実施状況

図5 CANDOの定着化状況 CANDOの適用率及び教育実施の推移について示す。



(a) 個人の習熟

(b) プロジェクトの習熟



(c) システム開発工程の作業比

図6 CANDOの適用による生産性の向上 生産性に関するCANDOの習熟曲線及び工程別工数分布を示す。

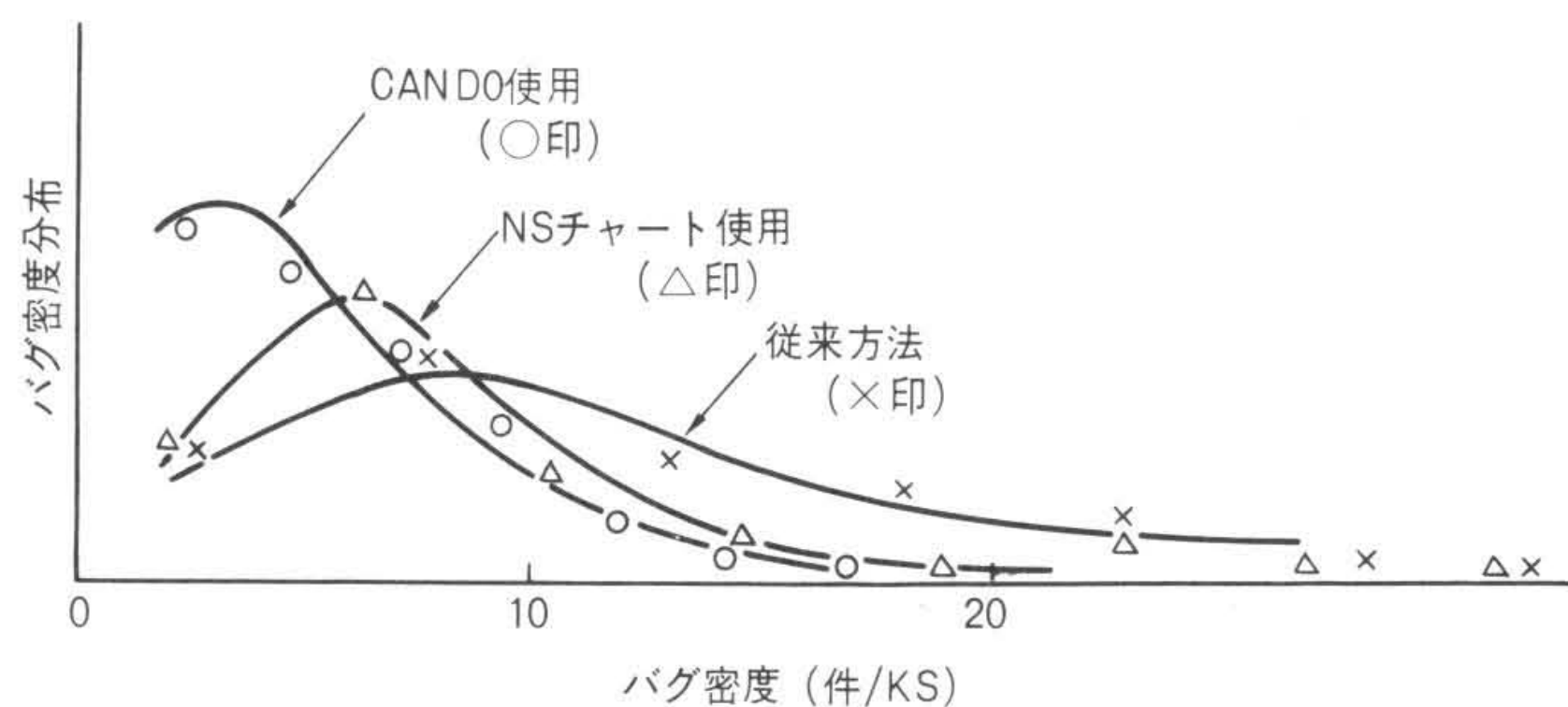


図7 単体テスト時のバグ密度分布図 単体テスト時のバグ密度をCANDO, NSチャート, 従来方式に分けて示す。

の分離が実現された。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。

5.4 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。特に構造化の徹底により、 $C_1=100\%$ を実現している。図7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、しかも集中していることが分かる。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。

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される。

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。

なお、本システムは、日立製作所の工場でプログラムプロダクト“EAGLE”として製品化が進められている。

参考文献

- 1) 片岡, 外: 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, 外: 構造化ソフト開発システムCANDOの開発及び評価, 昭和58年度情報処理学会第27回全国大会講演論文集, 521~522(昭和58-10)
- 3) 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)
- 4) G. J. Myers著(有澤訳): ソフトウェアの信頼性, 近代科学社(昭和52-10)