

トランザクション処理プログラムへの1構造的 アプローチ†

新 舎 隆 夫^{††} 塩 田 博 行^{††}
金 山 賢 一 郎^{†††} 春 名 宏 一^{††}

端末がインテリジェント化するにつれて、各種の端末群を集中的に制御、管理する TCE (Terminal Control Equipment) のソフトウェアは一層大規模化、複雑化してきた。

本論文では、専用端末固有の業務処理を行う OP (Operation Program) の生産性向上方法を論じる。最初に、OP の制御モデルを作成し OP の制御と処理を明確にする。次に、(1)制御と処理の分離、(2)システム規模に対する拡張性、(3)モジュール化の促進、(4)システム固有な制御情報の識別の4つの標準制御構造の基本条件の視点から OP を構造化する。最後に、これらの条件を満たすトランザクション処理プログラムへの1構造的アプローチ、SATC (Structured Analysis for Transaction Control) を提案する。

SATC は1種のより実用的で強力な決定表プログラムであり、3階層の標準制御構造とこれに裏付けられた機能設計の基本手順を与える。SATC は、トランザクション処理プログラムの機能設計と基本設計を単純化、形式化するので、このプログラムの標準設計手段であると考えられる。

最も代表的な専用端末である銀行の窓口装置に SATC を適用した結果、SATC はプロトタイプシステム開発時と比較して、トランザクション処理プログラムの全体開発工数を少なくとも約 15% 削減可能であることが判明した。

1. ま え が き

最近、オンライン・システム構成上の顕著な動向の1つに情報処理の分散化が挙げられる。この動向に対処する1つの方向として、端末機能の高度化—インテリジェント—化の要求が生まれ、広範囲の専用端末および汎用端末の分野で、端末のインテリジェント化が普及拡大している^{1),2)}。端末の高機能化、多様化に伴い、各種の端末群を集中的に制御、管理する TCE (Terminal Control Equipment) のソフトウェアは、一層大規模化、複雑化してきた。特に、専用端末固有の業務処理を行う OP (Operation Program) はこの傾向が顕著であり、ユーザ自身が OP の開発および管理を効率的に行うことができる、UPTS (User's Programmable Terminal System) の実現が重要な課題になった。

従来、OP の詳細設計以降のフェーズを支援する手段として、OP 向き簡易言語²⁾、既存の言語系 COBOL、RPG などがあったが、OP の機能設計、構造設計を支援する手段はほとんどなく重要な研究課題であった。筆者らは、あらかじめ OP が処理する情報の性

質の分析を通じて階層化された構造設計を行い、この構造に裏付けられた機能設計を行うという構造的アプローチにより、両者の効率向上を試みた³⁾。

この構造的アプローチは、OP が処理する情報の性質の分析を通じて OP の制御と処理を明確にし、次の4つの設計上の視点から OP を構造化する。(1)制御と処理の分離、(2)システム規模に対する拡張性、(3)モジュール化の促進、(4)制御情報のシステム固有性の識別。その結果、(1)、(2)、(3)各々が可能な制御構造であり、システムに固有な制御情報を対象にする制御構造という4つの基本条件を満たす制御構造が抽出できるとき、これを標準制御構造とよぶ。この標準制御構造を支援する制御プログラムは、固定形式の制御テーブルと固定制御ルーチンで構成可能である。これが1度確立されると、後続システム開発時の OP の基本設計は性能、メモリ量の容易な事前評価を行い、その結果を確認するだけでよい。また、機能設計は制御テーブル・エントリを設定するだけでよい。したがって、標準制御構造と機能設計の基本手順が与えられるプログラムは、機能設計および基本設計が単純化され形式化されるので、両者の開発工数を大幅に削減可能である。このとき、標準制御構造と基本手順の対はこれらが適用されるプログラムの標準設計手段であると考えられる。

本論文では、OP へ構造的アプローチを適用し、標準制御構造の基本条件を満たすトランザクション処理

† Structured Analysis for Transaction Control by TAKAO SHINSHA, HIROYUKI SHIOTA (Systems Development Laboratory, Hitachi Ltd.), KENICHIRO KANAYAMA (Asahi Work, Hitachi Ltd.), and KOICHI HARUNA (Systems Development Laboratory, Hitachi Ltd.).

†† (株)日立製作所システム開発研究所

††† (株)日立製作所旭工場

プログラムを抽出し、その標準設計手段としてSATC (Structured Analysis for Transaction Control) を提案する。SATC は結果として Decision Table (以後は決定表とよぶ) の適用である。決定表はシステム分析の1手段としてよく知られ実際面へ適用されている^{4),5)}。最近、構造化プログラミングの開発普及に伴い、決定表が有する、(1)設計条件のすべての組合せの保証、(2)すべての冗長性の除去、(3)すべての矛盾の検出などの長所が再評価され⁶⁾、実用的な決定表プログラムの実現方法が研究されている^{7),8)}。これに対して、筆者らの狙いはいくつかの特性を有する全体プログラムの内から、より実用的で強力な決定表プログラムを抽出することであり、その抽出条件は上述の標準制御構造の基本条件で与えられている。SATC はトランザクション処理プログラムの標準設計手段として決定表を適用している点が特徴であり、決定表の長所の範囲内で高信頼設計手段でもある。

本論文の構成は次のとおりである。2章で従来設計方法の問題点を指摘する。3章でOPへの構造的アプローチにより、OPの制御と処理を明確にし、標準制御構造の基本条件の視点からOPを構造化する。4章でSATCを提案し、標準制御構造と基本手順を詳述する。5章で制御方式、6章で適用例、7章で評価結果を述べる。

2. 従来設計方法の問題点

OPの機能に着目した従来設計方法の主な特徴として次の2つが挙げられる。(1)プログラム構成は、入出力、処理形態などによりOPの機能をいくつかに分け、各機能を1つのプログラムに対応づけた構成であり、(2)各プログラムは相互のインタフェース処理を支援する固有の制御部を有していた。プログラム構成例を図1に示す。その結果、制御部がOP内に散在してしまい、制御と処理の分離すらできず、標準制御構造の基本条件が満たされていなかった。これに起因して、OPが大規模化、複雑化するにつれて、OP開発工数が増大するという問題が生じた。

3. OPへの構造的アプローチ

OPが処理する情報の性質の分析を通じてOPの制御と処理を明確にし、標準制御構造の基本条件の視点からOPを構造化する。

3.1 OPの制御モデル

OPの制御モデルを図2に示す。この図に基づき大

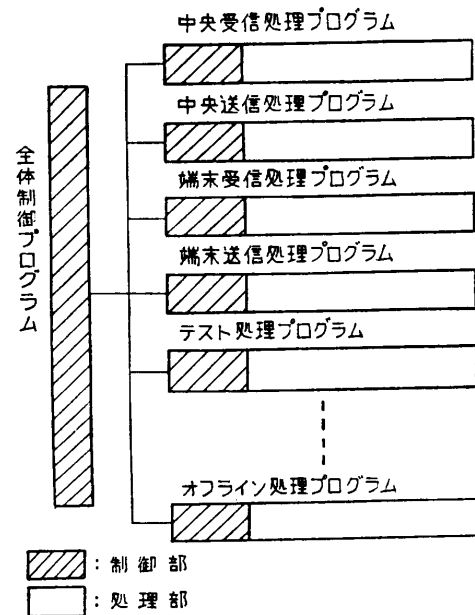


図1 プログラム構成例

Fig. 1 An example of an operation program structure.

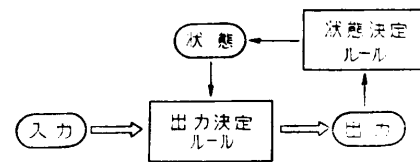


図2 制御モデル

Fig. 2 A control model.

規模端末システムのOPの制御と処理を明確にする。

(1) 入力: TCEは自ら処理を生成せず、中央と端末の間のインタフェース的な役割を果たす。すなわち、OPは中央、端末からのメッセージ受信をきっかけにして処理を開始するので、中央/端末種別とメッセージ種別の対は外部入力である。また、出力の一部であるI/O処理が異常終了したときに、状態に依存した障害処理が行われるので、I/O障害は内部入力とみなされる。

(2) 出力: 入力と状態により決定される処理ユニット(1処理の単位、I/O処理も含む)の集合とする。

(3) 状態: 取引シーケンスの進行状態を示す業務ステータス、制御ブロック内のTCEステータス、端末ステータスの3種類がある。

(4) 出力決定ルール: 入力と上記3種類のステータスの組合せにより出力を決定するルールである。

(5) 状態決定ルール: 次の入力に備えるため、出力

結果に従って状態を決定するルールである。業務ステータスの決定ルールは状態遷移方式であり、出力結果により3以上の状態のいずれかへの遷移を決定する。これに対して、TCEステータス、端末ステータス各々の決定ルールは状態保持方式であり、2状態のいずれかの保持を決定する。

この制御モデルにより、OPの制御とは出力決定ルールに基づく処理ユニットの決定および状態決定ルールに基づく状態の管理であり、OPの処理とは処理ユニットの集合であると考えることができる。

3.2 OPの構造化

上述のOPの制御モデルに標準制御構造の基本条件の視点を与えてOPを構造化する。

(1) 制御と処理の分離: OPの制御部は、外部入力に対する処理ユニットを決定する外部入力制御、内部入力に対する処理ユニットを決定する内部入力制御、状態遷移管理、状態保持管理で構成される。OPの処理部は処理ユニットの集合からなる通常処理である。この全体構成により制御と処理の分離が可能である。さらに、システム・プログラムがTCEステータスを管理し、端末がメッセージ送信時にいつも端末ステータスを送信するという設計方法を採用することにより、状態保持管理はOPから分離可能である。

(2) システム規模に対する拡張性: 業務ステータスの導入は中央とTCEの間の機能分担方式に依存する。一般に、大規模端末システムは中央負荷低減と機能拡張性を目的として業務ステータスを導入するが、中小規模端末システムはこれを導入しない。したがって、外部入力制御の内の業務ステータスに関する制御、内部入力制御、状態遷移管理はサブセット化される必要がある。内部入力制御全体がサブセット化される理由は、業務ステータスが導入されないとき、処理ユニットは処理結果として生じるI/O障害種別を容易に識別し、自らの判断だけで該当障害処理を行うことが可能であるからである。

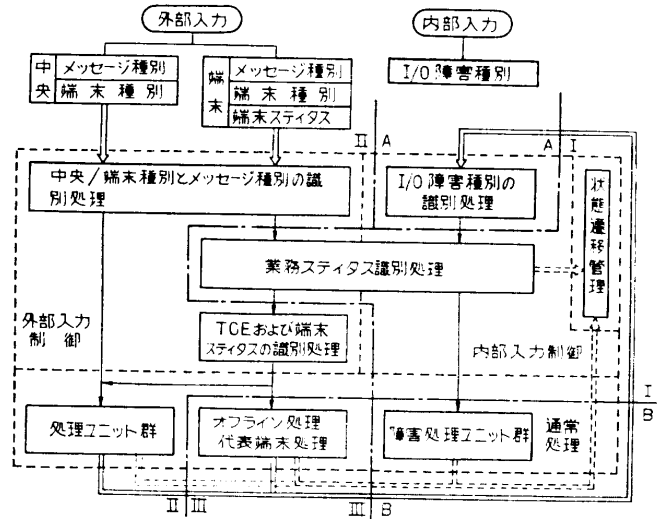
(3) モジュール化の促進: 通常処理は状態依存処理と入力データ依存処理に大別される。前者はトランザクション処理、障害処理であり、外部入力制御、内部入力制御により複数の適切なサイズの処理ユニットにモジュール化可能である。このとき、処理ユニットに関する処理の重複を防ぐために処理ユニットをさらにモジュール分割し、基本モジュールを設定すること

が可能である。業務ステータスが導入されないとき、障害処理ユニットは1以上の基本モジュールとして設定され、トランザクション処理ユニットに組込まれる。基本モジュール化を図るとき、処理ユニットの集合は、基本モジュール・シーケンス制御部と基本モジュールの集合からなる処理部で構成され、さらに制御と処理の分離が可能である。これに対して、後者に属するオフライン処理、代表端末処理(簡易ユティリティ)は外部入力制御、内部入力制御を行っても、モジュール化を促進しない。

(4) 制御情報の固有性の識別: OPの制御情報には、システムであらかじめ決められている固定制御情報と個々のシステムごとに設定される可変制御情報がある。端末種別、メッセージ種別、TCEおよび端末ステータスは前者に属し、業務ステータスは後者に属する。また、基本モジュール設定後、基本モジュール番号は固定情報であり、設計者が自由に扱うことができるので、本論文では一種の固定制御情報とみなす。

OPの構造化を図3に示す。OPの構造化により図1で示されるプログラム構成を再構成するとき、プログラム種別と各プログラムに対する標準制御構造の基本条件の適合性は次のとおりである。

(1) 業務ステータスに依存するトランザクション



- (注1)-----はOPの制御と処理の分離を示す
- (注2)-----はプログラム再構成時のプログラム種別を表す。ここで
 - I : 業務ステータスに依存するトランザクション処理プログラム
 - II : 業務ステータスに独立なトランザクション処理プログラム
 - III : 入力データ依存型処理プログラム
- (注3) 業務ステータスが導入されるとき、A、BはIに含まれ導入されないときII、IIIに含まれる

図3 OPの構造化

Fig. 3 Structuring of an operation program.

処理プログラム：制御情報の固有性が不適合である。
 しかし、これはサブセット化可能である。

- (2) 業務ステイタスに独立なトランザクション処理プログラム：すべての基本条件が適合する。
- (3) 入力データ依存処理プログラム：モジュール化の促進が不適合である。

4. SATC

SATC の狙いは、標準制御構造の基本条件をすべて満たす中小規模端末システムのトランザクション処理プログラムを対象にし、プログラム生産性向上を図ることである。SATC はこのプログラムの標準制御構造と機能設計の基本手順を提供する。

4.1 標準制御構造

SATC は PCC (Pattern Check Controller), MSC (Module Sequence Controller), 基本モジュールの集合からなる3階層の標準制御構造を与える。PCC は外部入力制御を支援し、外部入力に対して TCE および端末ステイタスの組合せにより処理ユニットを振分ける。MSC は基本モジュール・シーケンス制御を行う。この構造は決定表の1種である混合エントリ・テーブル方式⁴⁾の適用である。すなわち、この方式の条件、行動を各々ステイタス、基本モジュールに置換することにより、1メッセージ種別に対する OP サブ制御構造が得られる。各々のメッセージ種別に対する OP サブ制御構造をすべて統合したものが OP 制御構造である。OP 制御構造を図4に示す。

4.2 基本手順

SATC はまた、ステイタス・エントリと基本モジュール・エントリを設定する基本手順を与える。

- (1) まず、端末で行われるオペレーションのパターンを分類し、数種類の基本オペレーションを設定す

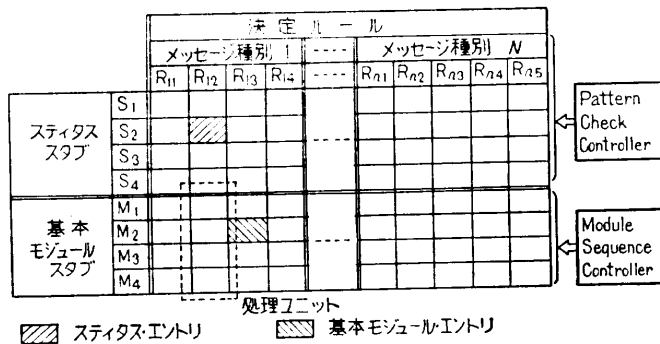


図4 オペレーション・プログラム制御構造

Fig. 4 The control structure for an operation program.

る。

- (2) 各基本オペレーションに対して、オペレーションにおける押下キーの制約が理解できるオペレーションフローを記述する。

(3) オペレーションフロー上で、キー押下により TCE へメッセージが送信される送信キーに注目する。各送信キーに対してあらかじめ設定されているステイタスの組合せを指定することにより、該当処理ユニットを決める。この操作により1基本オペレーションの処理ユニットの振分けが可能である。また、この操作をすべての基本オペレーションに対して行う。

(4) 各基本オペレーションの処理ユニット振分け結果を統合する。このとき、処理ユニットの同定および同一処理ユニットを振分けているステイタスの組合せの指定の統合が可能であれば合わせて行う。

(5) 障害メッセージのような非同期イベントに対しても、あらかじめ設定されているステイタスの組合せの指定により、別途処理ユニットを振分ける。

(6) 中央からの受信メッセージに対する処理ユニットを指定する。指定できる理由は、システム・デバッグ終了時に中央からの不当メッセージ受信、中央と TCE のステイタス不整合は生じないからである。

(7) 各処理ユニットが行う機能を機能モジュール・フローで記述する。

(8) 機能モジュールから基本業務モジュールを設定する。機能モジュールがそのまま基本業務モジュールになる場合と複数に分割される場合がある。これらに基本制御モジュールを付加して、各処理ユニットのモジュール・シーケンス・フローを記述する。

5. 制御方式

OP の標準制御構造を支援する制御プログラムは、2種類の制御ルーチンと5種類の制御テーブルで構成される。OP の制御方式を図5に示す。

5.1 コード化

SATC により得られる結果は次のようにコード化され、5種類の制御テーブルが得られる。

- (1) MCT (Message Check Table): これは端末用の MCT1 と中央用の MCT 2 の2種類がある。これらは端末種別、メッセージ種別と PCT (pattern Check Table) あるいは MST (Module Sequence Table) へのポインタの対応表である。

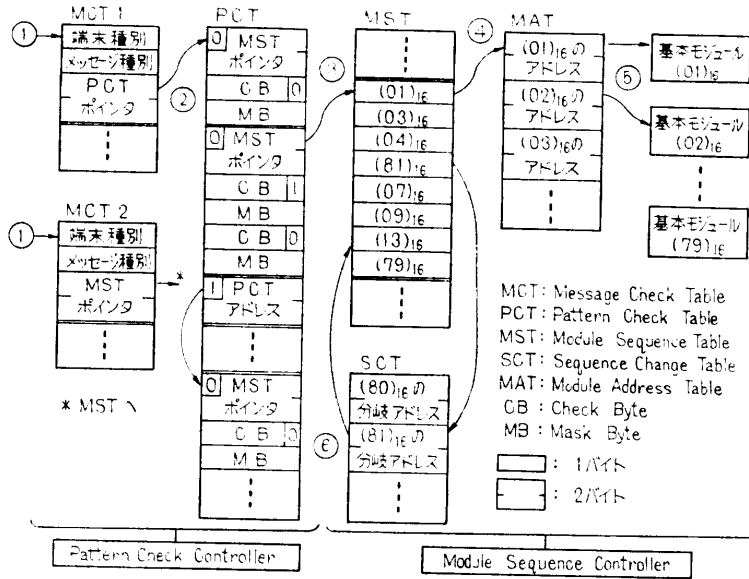


図 5 オペレーション・プログラムの制御方式

Fig. 5 The control method for an operation program.

(2) PCT: これは CB (Check Byte) と MB (Mask Byte) の対と MST へのポインタの対応表である。各ステータス・エントリは '1', '0', '-' のいずれかが指定されている。'1', '0' はあらかじめ定義されているステータス内容を示し, '-' は don't care を示す。CB の各ビットはステータスと 1 対 1 に対応づけられている。たとえば, ステータス集合 [S₁, S₂, ..., S₁₀] は第 1 CB [S₁, S₂, ..., S₇, C] と第 2 CB [S₈, S₉, S₁₀, /, ..., /, C] に対応づけられる。ここで, 'C' はチェインビットであり, C=1 は後続の CB があることを示す。'/' は未定義を示す。S₈=S₉=S₁₀='-' であるとき, 第 1 CB は C=0 になり第 2 CB は生成されない。'-' と '/' は 0 にコード化される。'-' に対応する 0 と '0' を識別するために MB が必要であり, '-' は MB で 1 にコード化される。ステータス数の増加はチェインビットの利用により容易に対処可能である。また, MST へのポインタの最上位ビットはアドレス意味フラグを示す。これが 1 のとき, このポインタは PCT アドレスを指す。これは異なる端末種別あるいはメッセージ種別に対する共通のチェック条件と処理ユニットの指定に関して, テーブルの共有化を図るためである

(3) MST: これは基本モジュールの処理手順を示す。一連の基本モジュール・エントリの集合が 1 つの処理ユニットを表わす。各エントリはその値が (80)₁₆ 未満のとき MAT (Module Address Table) を

指し, それ以上のとき SCT (Sequence Change Table) を指す。

(4) MAT: これは基本モジュールのアドレス・テーブルである。

(5) SCT: これは変更すべきモジュール・シーケンス・カウンタの値がセットされているテーブルである。

5.2 制御方式

OP の制御方式を図 5 中の番号にそって以下に説明する。(1) PCC はシステム・プログラムよりメッセージ受信を報告される。(2) 端末からのメッセージ受信時に, PCC は MCT 1 をサーチし PCT へのポインタを得る。また, 中央からメッセージ受信時に, PCC は MCT 2 をサーチし MST へのポインタを得る。

(3) PCC は SB (Status Byte: その時点のステータス内容がセットされた CB と同形式のバイト) を作成し, PCT のチェック条件と比較する。すべての i について,

$$[SB(i)] \oplus [CB(i)] \oplus [MB(i)] = (00)_{16}$$

$$(i=1, 2, \dots) \oplus: \text{Exclusive Or}$$

が成立するとき, それはチェック条件との一致を示す。PCC はチェック条件が一致したところの MST へのポインタを得る。(4) MSC は MST より該当基本モジュールのアドレスを得る。(5) MSC はこのアドレスに基づき基本モジュールをコールする。1 つの基本モジュールの処理が終了すると, 制御が MSC へ戻り MST のポインタが 1 加算され, 以下(4), (5) が繰返される。(6) 基本モジュールの処理結果により複数の戻り方があるときに, MST のポインタの変更が SCT により可能である。

PCC の決定表変換アルゴリズムは Kirk の Simple Rule Mask Technique⁴⁾ を基礎にして, CB, アドレス意味フラグの導入により, メモリ量削減を図っている

表 1 端末からの受信メッセージ例

Table 1 An example of messages from terminals.

種別	メッセージの意味
再表示	再表示キー押下時 TCE へ送信。入力画面時受信データをキューし, 後に続くガイド画面, 入力キュー画面を端末へ送信。出力画面時, 後に続く出力キュー画面を端末へ送信。
完了	完了キー押下時 TCE へ送信。入力画面時入力キュー画面, 単独画面を中央へ送信。出力画面時 QUIT を中央へ送信。
OPIM 2	端末が準備完了状態になると TCE へ送信。端末からの受信メッセージを中央へ送信。

ことが特徴である。また、MSC はモジュール制御でよく使用される方法である

6. 適用例

本論文の範囲内で、最も代表的な銀行の窓口装置⁹⁾に SATC を適用した結果を以下に述べる。

端末からの受信メッセージは 11 種類であり、ステ

イタスは 10 種類である。これらに基づいて振分けられた処理ユニット総数は 66 個であり、処理ユニット同定後は 29 種類である。端末からの受信メッセージ例を表 1 に、ステイタスを表 2 に、これに関する処理ユニットを表 3 に示す。また、中央からの OP に関する受信メッセージは 22 種類であり、各メッセージに対する処理ユニットが存在する。

表 2 ステイタス一覧表

Table 2 A status table.

略号	名称	分類	内容	1	0
S ₁	初期設定	TCE	初期設定有はTCEが中央から店番情報、日付情報を受信し、オンライン取引が可能であることを示す。	有	無
S ₂	タスクモード	TCE	タスクモードはオンラインかオフラインのいずれかである。	オンライン	オフライン
S ₃	TCEループ	端末	TCEループ有は端末にテストキーが挿入されていることを示す。	有	無
S ₄	オペレータキーの有無	端末	オペレータキー有は端末にオペレータキーが挿入されていることを示す。	有	無
S ₅	オペレータキーの種類	端末	オペレータキーは通常(テラー用)か特殊(役付者用)のいずれかである。	特殊	通常
S ₆	端末モード	端末	端末モードはオンライン系(オンライン、練習、再入)かオフラインのいずれかである。	オンライン系	オフライン
S ₇	送信キュー	TCE	送信キュー有は中央へ送信する複数画面入力があることを示す。	有	無
S ₈	受信キュー	TCE	受信キュー有は端末へ送信する複数画面出力があることを示す。	有	無
S ₉	モニタキューNO	TCE	モニタキューNOが0のとき、複数画面入力時のモニタ印字終了を示し、0以外のときはモニタ印字中を示す。	0以外	0
S ₁₀	入力データの有無	端末	入力データ有は該当入力メッセージにキー入力データがあることを示す。	有	無

表 3 処理ユニット例

Table 3 An example of processing units.

項目	ユニット処理決定ルール																													
	再表示								完了												OPIM2									
ステイタス・スタブ	S ₁	1	1	1	1	1	0	-	-	-	1	1	1	1	1	1	0	-	-	-	-	-	-	1	1	1	1	0	処理ユニット・スタブ	準備未表示(1)処理
	S ₂	1	1	1	0	0	-	-	-	-	1	1	1	1	0	0	-	-	-	1	1	0	0	1	0	0	0	-		準備未表示(1)処理
	S ₃	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	1	-	-	-	-	-		準備未表示(1)処理
	S ₄	1	1	1	1	1	1	1	0	-	1	1	1	1	1	1	1	0	-	-	-	-	-	-	-	-	-	-		準備未表示(1)処理
	S ₅	0	0	0	0	0	0	1	-	-	0	0	0	0	0	0	0	1	-	-	-	-	-	-	-	-	-	-		準備未表示(1)処理
	S ₆	1	1	0	1	0	-	-	-	-	1	1	1	0	1	0	-	-	-	1	0	1	0	1	0	1	0	-		準備未表示(1)処理
	S ₇	-	-	-	-	-	-	-	-	-	1	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		準備未表示(1)処理
	S ₈	0	1	-	-	-	-	-	-	-	1	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		準備未表示(1)処理
	S ₉	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		準備未表示(1)処理
	S ₁₀	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-		準備未表示(1)処理

(注 1) S₁, ..., S₁₀ は表 2 を参照のこと。(注 2) *印はトランザクション処理に属さない

トランザクション処理の占有メモリ量は OP 全体の約 50% である。PCC, MSC 各々の占有メモリ量は約 350, 70 B (Byte), MCT, PCT のそれは約 480 B, MST, MAT および SCT のそれは約 850 B である。これらの制御プログラム全体の占有メモリ量は約 1.8 KB であり, OP のプロシジャ部全体の約 9% に相当する。

機能フローとモジュール・シーケンス・フロー例を 図 6 に示す。機能性を保ちかつ独立性の高い基本モジュールを設定するために, OP の共通処理部分を基本モジュールより下位レベルのサブルーチンにすることが必要である。サブルーチン群を抽出後, トランザクション処理の範ちゅうに属する処理ユニットを対象にしてモジュール化を行った結果, 91 個の基本モジュールが得られた。これらの平均サイズは約 37 B である。基本モジュール・サイズの分布を 図 7 に示す。

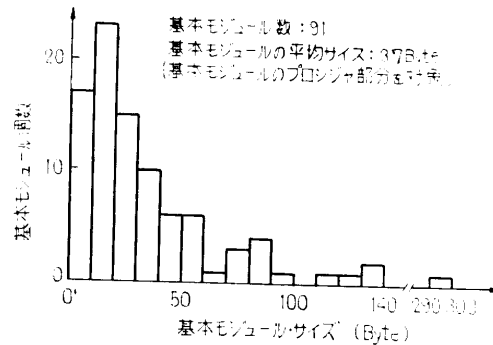


図 7 基本モジュール・サイズの分布
Fig. 7 A distribution of basic module size.

7. 評価結果

制御方式のオーバーヘッドが許容範囲内であることの確認, トランザクション処理プログラム開発工数削減率の 2 つの視点から SATC の評価を行う。

7.1 制御方式の評価

PCC, MSC 各々 1 回の実行ステップ数は約 300, 20 S (Step) である。1 取引当りの PCC, MSC 各々の実行回数は約 4, 14 回である。これらは全体実行ステップ数 (約 20 KS/取引) の 8% 以下であり無視できる。

制御プログラム全体の占有メモリ量は, ステップ数に換算 (3B/S) すると約 600 S である。MST には約 400 個の基本モジュールが登録されており, これを Branch and Link 命令 (4B) で実現すると, この部分だけで約 1.6 KB が必要である。したがって, 600 S という値は 88 個の処理振分けと 91 種類の基本モジュールに関するモジュール・シーケンスをプログラムロジックで制御するより小さくなる。

7.2 開発工数の評価

SATC の主な長所は次のとおりである。(1) SATC の基本手順により標準制御構造に裏付けられた機能設計が可能であり, その効率向上が図れる。(2) 標準制御構造の確立により, 性能, メモリ量の事前評価が容易であり, 基本設計はこれらの評価結果を確認するだけでよい。(3) 機能性を保ちかつ独立性の高い基本モジュールの設定により, 後続システム開発時にこれらの利用率が増大し, 新規開発の対象となる基本モジュール数を削減できる。(4) 仕様変更, 機能拡張は制御テーブルの一部および基本モジュールの変更, 追加で容易に対処可能である。

SATC の欠点は, 段階的なプログラム開発時に生

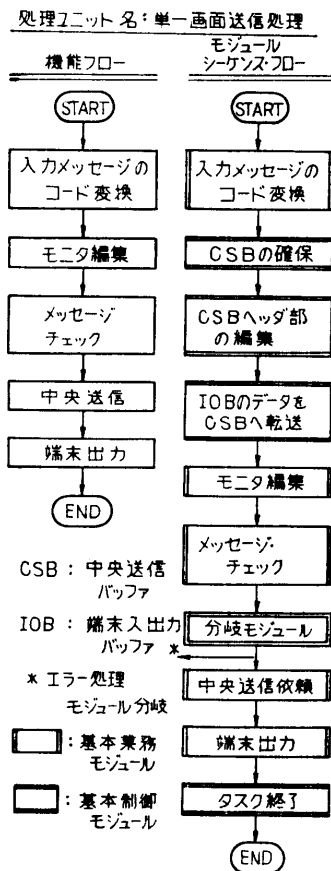


図 6 機能フローとモジュール・シーケンス・フロー例
Fig. 6 An example of functional flow and module sequence flow.

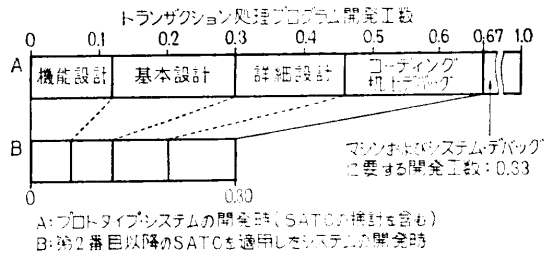


図8 トランザクション処理プログラム開発工数の比較

Fig. 8 A comparison with regard to manpower estimate in developing transaction processing programs.

ずる制御テーブルの保守、基本モジュールの同定などのプログラムの保守作業が不可避であることである。これはプログラムの共有化を図って開発するときの基本問題に起因している。また、基本モジュールの明確な設定基準および検索方法も依然基本問題として残されている。

プロトタイプ・システム開発時とSATCを適用したシステム開発時のトランザクション処理プログラム開発工数の比較を図8に示す。この図は後者が前者と比較して約37%開発工数削減可能であることを示している。開発担当者の経験の蓄積を考慮しても、SATCは上述の長所によりこの値の少なくとも40%に寄与していると考えられる。

8. む す び

専用端末のトランザクション処理プログラムへの1構造的アプローチ、SATCを提案し、窓口装置を対象にしてその評価を行った。今後、(1)業務ステイタスを導入したときの高信頼標準制御構造の確立、(2)これを基礎にして、種々の状況下での容易なプログラ

ム開発を支援するツールの開発などを通じて、UPTSの実現を図ることが重要な課題であると考えられる。

謝辞 本研究の機会を与えて頂いた、システム開発研究所の三浦武雄博士および当社旭工場の笹倉邦彦部長(当時)、また、制御方式の共同検討をして頂いた、旭工場の佐々木哲允氏および当社ソフトウェア工場の西川明宏氏に深謝いたします。

参 考 文 献

- 1) 浦木ほか：端末システムの動向，日立評論，Vol. 59, No. 2 (Feb. 1977).
- 2) 田沼ほか：機能分散化とインテリジェントターミナル，情報処理，Vol. 18, No. 4 (Apr. 1977).
- 3) 新舎：拡張性のある端末制御ソフトウェア構造に裏付けられた機能記述方式，昭53年度電子通信学会総合全国大会。
- 4) Pollack et al.: Decision Table: Theory and Practice, John Wiley & Sons (1971).
- 5) Low: Programming by Questionnaire: An Effective way to Use Decision Tables, C. ACM, Vol. 16, No. 5 (May 1973).
- 6) Gildersleeve: The Dark Side of Structured Programming, Datamation, Vol. 21, No. 11 (Nov. 1975).
- 7) Lew, et al.: Decision Table Programming and Reliability, Proc. 2nd Int. Conf. on Software Eng. (1976).
- 8) Lew: Optimal Conversion of Extended-Entry Decision Tables with General Cost Criteria, C. ACM, Vol. 21, No. 4 (Apr. 1978).
- 9) 太田ほか：金融業における端末システム，日立評論，Vol. 59, No. 7 (July 1977),

(昭和54年8月1日受付)

(昭和55年3月21日採録)