U.D.C. [681.323.06:621.395.345].001.31

電子交換機の機能分散ソフトウェア方式

Functional Distributed Modular Software Structure for ESS

桧山邦夫* Kunio Hiyama
水原　登* Noboru Mizuhara
鈴木太平** Taihei Suzuki
河野善弥*** Zenya Kôno

電話の普及とともに日常必須な道具となるのに従い，電話の効率向上の要望が高まり，そこで種々の高度な機能の導入が図られている。その結果，交換ソフトウェアは大規模化，複雑化の傾向をたどり，それに対応した新しいソフトウェア方式の開発が急務となってきている。

このような背景から，「分散形状態遷移方式」を開発し導入した。本方式は論理的な交換機能をとらえ，複数の仮想機械に分割すると同時に階層分割化して，各々を状態遷移図で図示化し，それに基づいたトップダウンなプログラム構造化を図った2次元階層構造をもつ方式である。これにより，将来多様化する通信サービスへの対応，更に分散制御方式への対応も可能とし，ソフトウェアの信頼性，生産性及び機能拡張性を高めた。

1 緒　言

近年，電話が日常生活の必須機能になるに従い，より効率の良い電話サービスが望まれ，電子交換機は，その蓄積プログラム制御の特徴を生かし，それらの実現を図ってきている。その結果，交換ソフトウェアは大規模かつ複雑化の傾向を強めている[1)]。

一方，従来はプロセッサを中心とする処理装置のコストが高いために，規模に応じて異なる機種を用いざるを得ず，その結果，ソフトウェアを規模ごとに独立に開発をしていた。しかし近年，高性能なマイクロコンピュータの出現により，大幅な低コストで処理装置が実現できるようになった。そこで，複数のマイクロコンピュータを規模に応じてビルディングブロック方式により構成することが，分散制御方式を導入することにより可能となった。その結果，同一方式で幅広い規模の拡張性をもつだけでなく，更に同一アーキテクチャ化されることにより，ソフトウェアの統一が可能となった。

一方，交換方式も従来のアナログ空間分割方式からディジタル時分割方式に変化しつつあって，それに伴う将来の動向を見越した新しいソフトウェア制御方式とする必要を生じている。

更に，今後とも通信の効率化をねらった新しいサービス機能の実現が強く望まれるであろう。

以上の背景のもとに，それらに対応できる新しい機能分散ソフトウェア方式を「分散形状態遷移方式」として新たに開発した。以下にその概要を述べる。

2 開発のねらい

2.1 将来への拡張性

交換システムとして，将来への高い拡張性をもっておくために，ソフトウェアシステムにも高い拡張性が要求される。

将来への動向として，大きな点としてはISDN(Integrated Services Digital Network)化があろう。これはディジタル時分割方式化に伴い，音声のディジタル通信に伴せて計算機データ情報，更に画像情報などの多元情報を一括して扱う通信システム化である。この変革から，一つは交換スイッチの多元ネットワーク化方向と，もう一つは多様な高度端末の接続可能化方向とが考えられるであろう。すなわち，交換接続機能の多元化だけでなく分散化，更には広域ネットワークとしての効率の良い通信システム化が要求されるであろう。一方，端末は従来の電話機だけでなく高度なインテリジェンスをもつ多様な端末が出現し，使用者の使い勝手の向上も要求されるであろう。これらの動向は一斉には生せずむしろ漸進的に進展していくものと思われる。

そこで，これらの動向に対応するためには，機能モジュール化を徹底して図り，進展に対応したモジュール機能の部分的変更だけで対応可能としていく必要がある。以上から，少なくとも端末種別ごとの機能分割と，ネットワークスイッチング機能の独立化とがなされる必要があると考える。

2.2 サービス機能の拡張性

既に現在の交換サービス機能も非常に豊富な機能をもっているが，更に通信の効率を向上させるニーズも強く，今後とも多様な新サービス機能の拡充を要するであろう。

また，これらサービス機能は，用途，顧客ごとに一般に異なるものも多くある。そこで，これらの多様なサービス機能を実現するためには，実現手段であるソフトウェアを機能との対応が明確な多数のモジュールに分割し，それらの組合せによる方法が必須となる。この方法を用いれば，共通な部分の明確化も図れ，新機能の実現にもわずかなプログラム開発で対応できるであろう。

一方，ソフトウェアは一般に内容がVisible(可視的)ではなく，第三者に理解しにくいものになりがちである。そこで，これを避けるためには，一つは実現すべき機能仕様をすべて図示できるようにすること，もう一つはそれに基づいたトップダウンな設計を行なうことが必須となる。この基本的方法が実現できればソフトウェアは交換機能に対応した構造となり，前述の目的が達成できる。また，設計手法の標準化が図られ，設計内容の信頼性の向上が得られ，更に生産性の向上が大きく期待できる。そこで以下に述べるような新ソフトウェア方式を開発し，その方法の具体的な実現化を図った。

* 日立製作所システム開発研究所　** 日立製作所戸塚工場　*** 日立製作所戸塚工場 工学博士

3 機能仕様の記述方式

3.1 仮想交換機と仮想機械への分割

前述のように実現すべき交換機能の仕様を明確に記述することが，まず第一に必要である。交換機能のとらえ方として，**図1**に示すように交換機をブラックボックスとしたときの外部から見た動きでとらえる方法がある。これは実現手段に依存しない機能表現であり，仮想的な交換機の機能仕様と言えるであろう。

外部インタフェースの代表例として電話機を採り上げると，**図1**に示すように発信手順と着信手順とが独立にある。その手順により多様な交換サービスを選択することもできる。交換機はこの手順を実現し，そのとおりに制御する必要があり，この手順は機能仕様となる。一方，電話機以外にも外部インタフェースは，交換台，他の交換機との回線などがある。それらの制御手順は似たものではあるが異なっており，各々独立な手順である。そこで，このような制御手順をもつ仮想的な順序機械を想定する。この仮想機械は制御手順種別ごとに独立に設ける。すなわち外部インタフェースの種別ごとに仮想機械を設ける。これらの仮想機械間を，仮想的なスイッチ機械で接続されるとし，これにより端末間の通話がなされるとする。

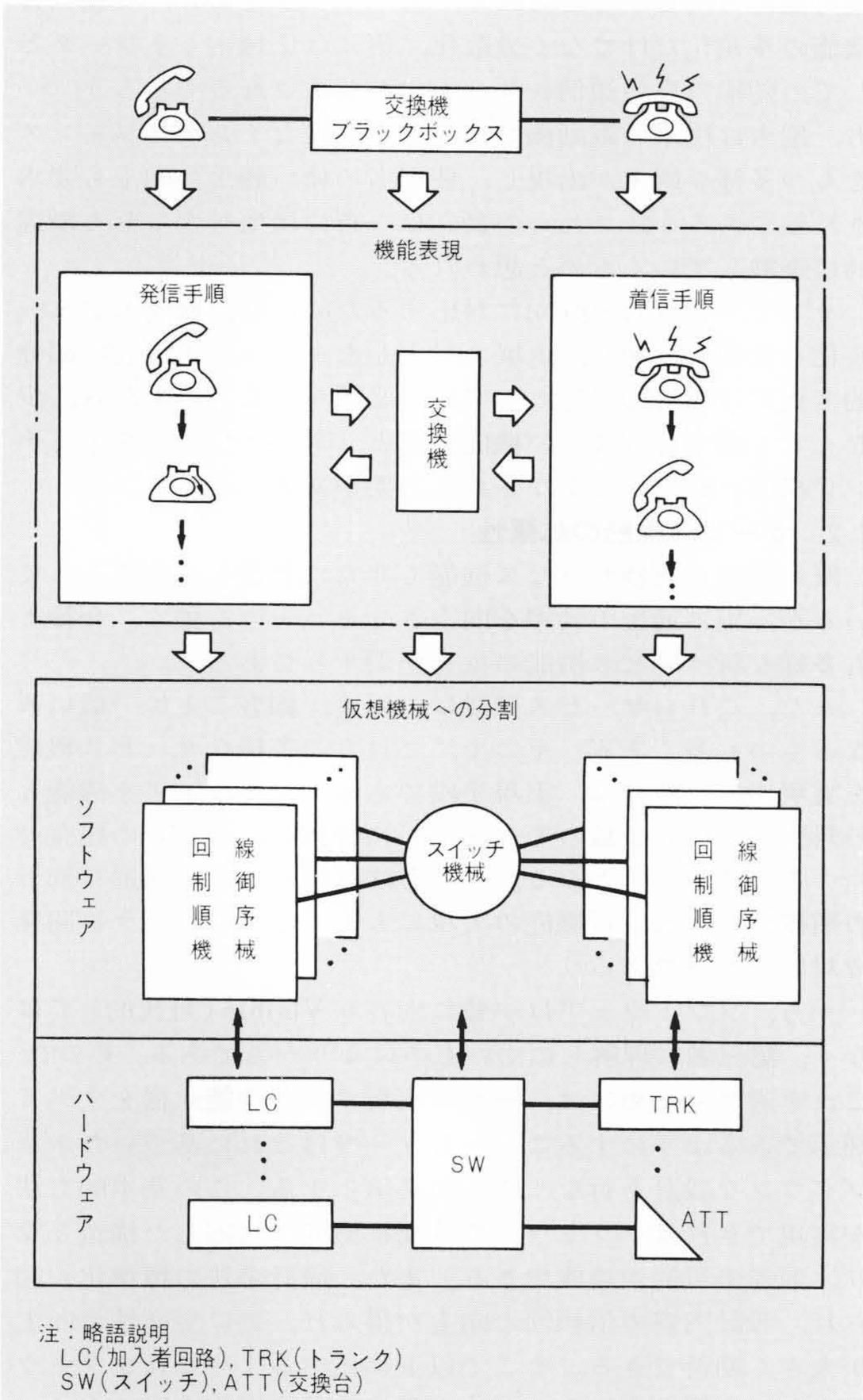

図1 仮想機械への分割 交換機をブラックボックスとし，外部から見た機能(仮想交換機)を手順としてとらえ，その手順を実現するソフトウェアを仮想的な順序機械として分割する。

表1 仮想順序機械例 ソフトウェア機能を，複数の仮想順序機械へ分割した具体例を示す。

名称及びサブマシン	制御対象
加入者回線制御 1．発信サブマシン 2．着信サブマシン 3．多者通話サブマシン	加入者回線
交換機間回線制御 1．発信サブマシン 2．着信サブマシン	トランク
交換台制御 1．交換台サブマシン 2．仮想扱い者サブマシン	交換台

この想定は，**図1**に示すように交換機の実際のハードウェア構成との対応もとりやすい。仮想スイッチ機械はSW(スイッチ)，仮想順序機械はLC(加入者回路)，TRK(回線トランク)，ATT(交換台)と対応し，その制御をつかさどる位置づけとなる。

この仮想機械への分割により，将来動向への漸進的対応を容易となし得よう。新しい端末の出現に対しても，その手順制御をもつ仮想機械の導入により，またスイッチのネットワーク化に対しても，仮想スイッチ機械の取替えだけにより対応が可能である。

仮想機械の例を**表1**に示す。大きくは，加入者回線，交換機間回線及び交換台の3種のインタフェースの仮想機械に分割している。各々の仮想機械の中に，更に手順の相違から複数のサブマシンを設けている。例えば，加入者回線，すなわち電話機の制御手順では前述のような発信と着信，それ以外に多者通話サブマシンを設けている。多者通話は，同時に複数の相手と通話ができるサービス機能で，そのときの制御手順は他と異なるので独立化を図っている。交換機間回線も同様に発信と着信とのサブマシンに分割している。一方，交換台の制御は，交換手の手順を制御する交換台サブマシン以外に仮想的なサブマシンを導入している。これは，仮想的な交換手によるサービスとも言える機能の実現のためであり，例えば，キャンプオンサービスなどがある。キャンプオンサービスは，話中に遭遇したときに，相手が空きしだい呼び出してくれる機能で，仮想的な交換手が代わって監視及び呼出し，接続するとみなせる。この概念により，手順の複雑化が避けられている。

3.2 状態遷移図

各仮想順序機械は，一つの安定状態に対して入力を与えることにより次の安定状態に遷移する状態遷移制御とみなすことができる。このような状態遷移を表現する方式は一般にも広く行なわれており，**図2**に示すペトリネットもその一例である。

一方，交換機の仕様記述方式として，CCITT[2](国際電信電話諮問委員会)でも状態遷移図による図示表現を勧告しており，その表現方法は，遷移処理に必要な処理機能を更に記述できるフローチャート流儀を加味したものである。そこで，

その勧告に基づきそのサブセットを，**図2**に示すように用いている。状態遷移の契機を示すイベント入力，遷移処理に必要とする他への出力及び遷移処理の分岐を示す分析の3種だけの記述を状態間の遷移中に記述できるようにしている。

3.3　機能の階層分割

複数の仮想機械による仮想交換機の機能を，実交換機ハードウェアにより実現するためには，更にその手順を交換機ハードウェアとのソフトウェアインタフェース規約まで分解する必要がある。

そこで，交換機能を仮想交換機から実交換機ハードウェアに至るまで**図3**に示すような5階層に分割する。最下位のレベル0に実交換機ハードウェアを置き，最上位のレベル4に仮想機械群から成る仮想交換機のサービス手順を置く。その間を3分割し，レベル1には実交換機ハードウェアとのソフトウェアインタフェースであるアクセス手段を密着させて設ける。レベル2では，ハードウェア回路での信号作成及び受信のために必要な制御手順を規定する。レベル3では，端末，

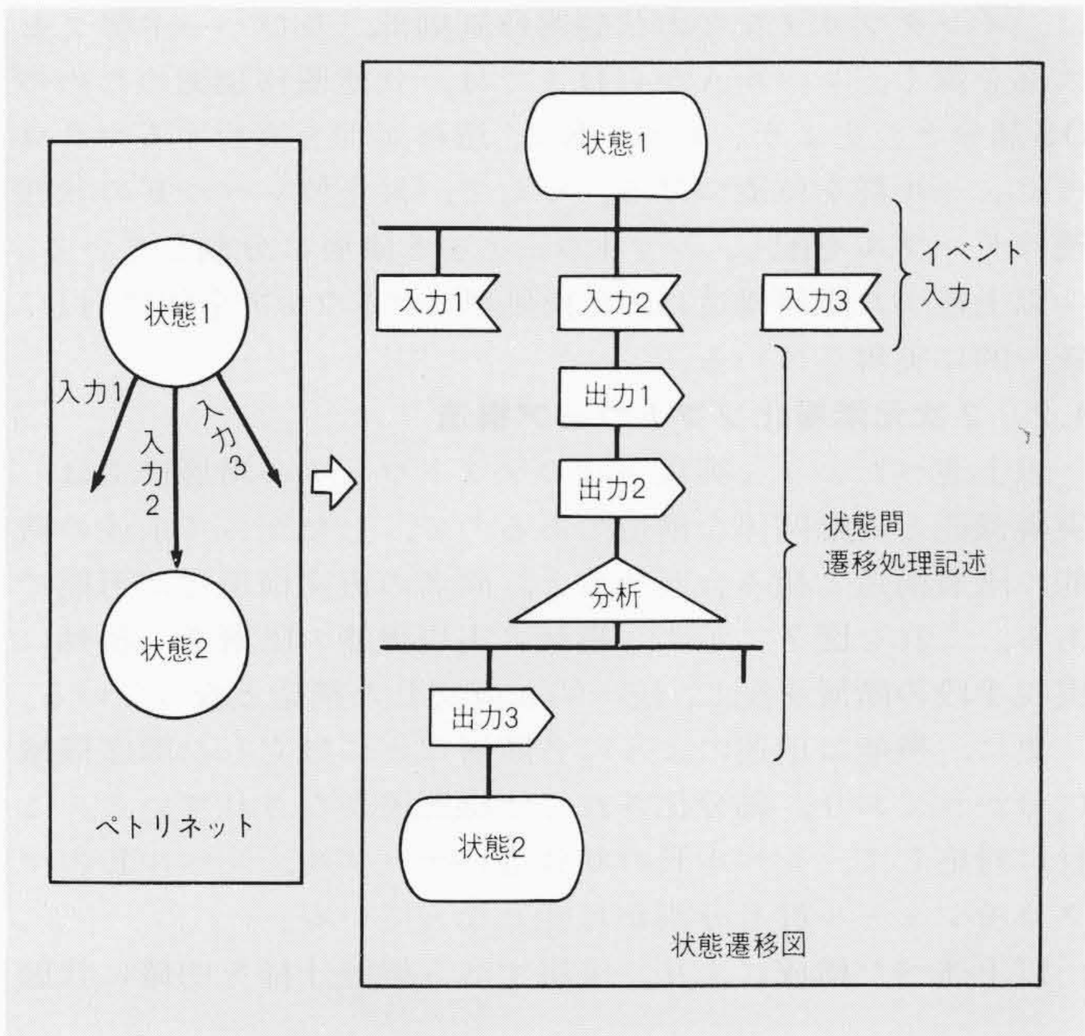

図2　状態遷移図　状態遷移図とペトリネットを比べると，状態間の遷移処理を明示している点が異なる。

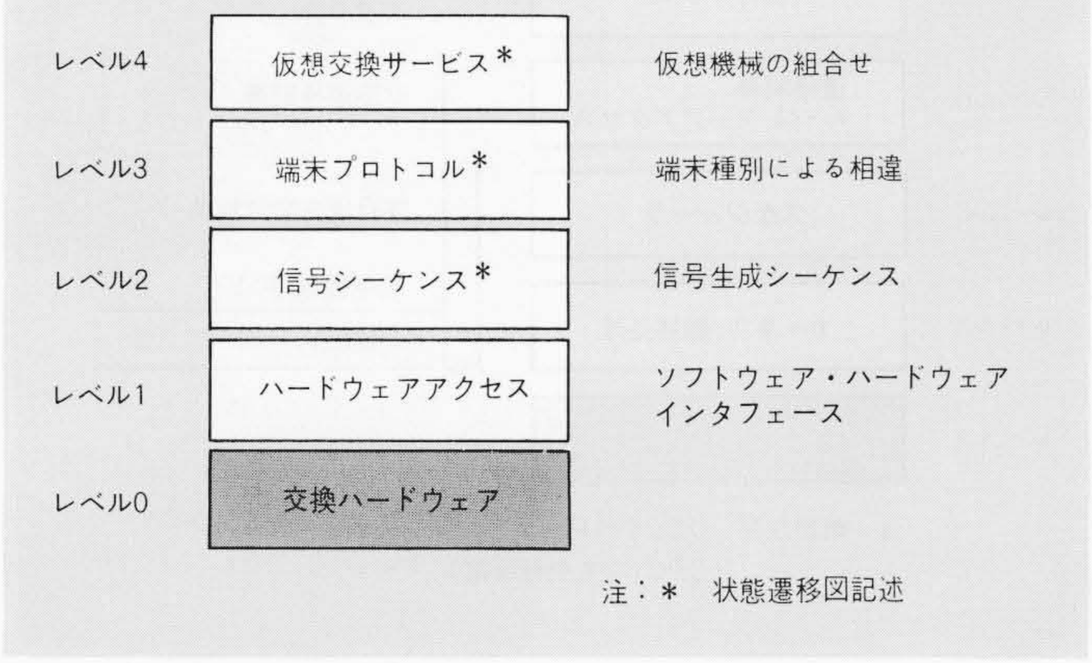

図3　交換機能の階層分割　交換機ハードウェアを中心に，交換機能の実現のためのソフトウェア機能を目的別に階層分割する。

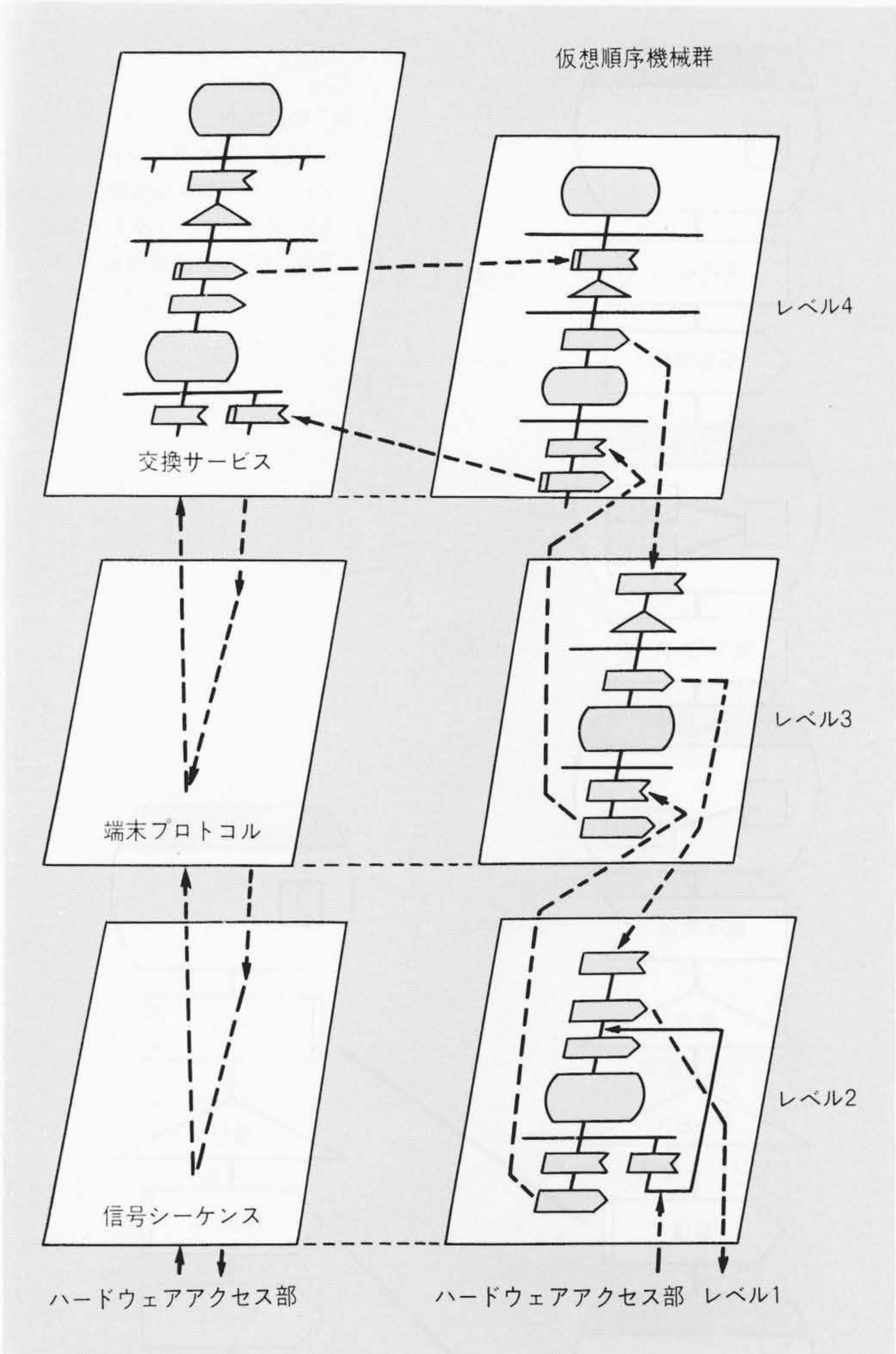

図4　分散形状態遷移図　仮想機械，階層ごとに独立した状態遷移図とし，各状態遷移図間は，出力→入力により連結動作を行なう。

回線の種別により多様性をもつ信号の組合せシーケンスの相違を規定する。これらの階層のうち，上位レベルの2，3，4は状態遷移図で記述された仮想機械となっている。

このような階層分割化とそれらの状態遷移図記述により，全体の機能仕様を統一した記述方法により可視化できる。更に，機能変更，拡張に対しても対応部分が局所化でき，内容が明確なため対処が容易である。例えば，サービス仕様の変更はレベル4，また特に多様性の多い交換機間プロトコルの拡張はレベル3，信号作成，受信手順の変更はレベル2，ハードウェアのアクセス法の変更はレベル1と，内容に応じて変更箇所の局所化を可能とする。

レベル2，3，4での各々独立した状態遷移図で記述された仮想機械間の制御は，**図4**に示すように状態遷移中の出力をすべて隣接する階層の仮想機械への入力としており，これにより，一貫した階層間の制御動作を可能としている。レベル4の交換サービス階層では，更に仮想機械間あるいはサブマシン間での交信を行ない，仮想機械間の相互制御を行なっている。このように，各々独立した仮想機械間を入出力により有機的に結合している。

状態遷移図の具体例を**図5**に示す。同図は，レベル4での電話機の発信と着信のサブマシン間の状態遷移図で，発呼から応答までの部分を示している。発呼は，レベル0のハードウェアをレベル1のアクセス部を通じてレベル2で検出し，

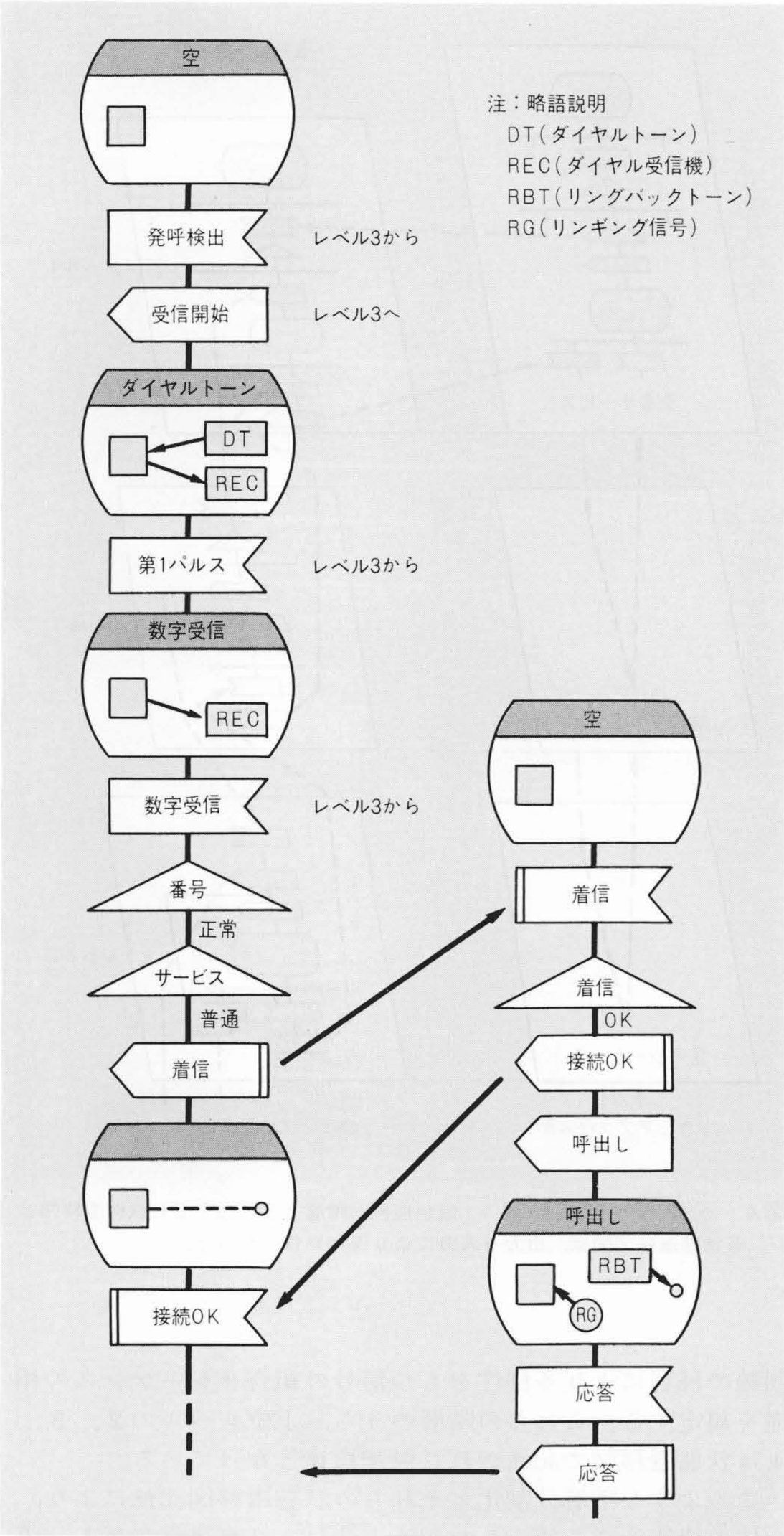

図5 状態遷移図例(レベル4) 電話機の発信から呼出しまでの交換サービス(レベル4)の状態遷移図例を示す。仮想機械間，下位レベル間の信号送受を明記し，これらをマクロとする。

レベル3でその確認シーケンスを経て，同図のレベル4に発呼検出のイベント入力として伝えられる。それにより，レベル4は遷移処理を開始し，レベル3にダイヤル受信の指示を与える。以下同様にダイヤルトーン出力，ダイヤル受信の状態を経て，番号分析で受信ダイヤル数字のチェック，サービス分析でダイヤル内容によるサービス種別の決定が行なわれる。それに従い，着信サブマシンに着信出力により起動をかける。以下，発信サブマシン，着信サブマシン間での信号のやりとりにより，同図のレベル4の状態遷移制御が進むと同時に，レベル4からの出力指示によりレベル3，レベル2の制御シーケンスが動作し，レベル1を経由してレベル0の実ハードウェアが制御される。このように，レベル4が全体の制御の主導権をもっている。

4 ソフトウェア構造

4.1 実現手段の階層構造化

交換機能を階層分割及び仮想機械に分割し，状態遷移図によりすべて図示化が図れるので，次にこれに基づくトップダウンなソフトウェア構造化が必要である。

状態遷移制御の実現方式として，状態遷移処理をマクロ言語のシーケンスとし，その解釈実行方式（インタプリティブ方式）による方法があり，従来の交換ソフトウェアでも行なっている。そこで，今回はこの方式を統一的かつ全体的に採り入れることとした。その結果，実現手段のソフトウェア構造は，**図6**に示すように5階層に統一的に分割できる。

状態遷移図の記述内容を，マクロ言語のシーケンステーブル化し，それを最上位に位置づけする。最下位のレベルAには，実行手段ハードウェアであるCPU(中央処理装置)を配する。その間を4分割し，レベルBにはCPUに密着した割込処理部，障害処理部を置き，レベルCにはCPUの実行優先制御を行なうスケジューラを設ける。レベルB，Cは一般的なOS(オペレーティングシステム)機能の一部である。レベルDには，インタプリタなどの状態遷移制御部，及びハードアクセス部を置く。レベルAからDまでは，状態遷移制御のためのOS部分と言えよう。レベルEに，遷移処理を実行するマクロモジュール群を位置づける。そして，最上位レベルFの状態遷移テーブルを配し，ソフトウェアを5階層に分割している。

以上述べた階層構造は，交換処理ソフトウェア全体に対し，統一的に実現している。

4.2 2次元階層化ソフトウェア構造

以上述べたような実現手段のソフトウェアの階層構造は，実現機能とは無関係な構造であるので，したがって前述の機能の階層構造と組み合わせると，両者の直交構造化が可能である。これを**図7**に示す。縦軸に実現機能の階層を，横軸に実現手段の階層を配し，統一的な直交した構造となっている。

更に，機能は前述のように各階層ごとに独立した順序機械に分かれており，細分化されて仕様記述がなされている。これに対応して，レベルFの状態遷移テーブル，レベルEのマクロモジュール群も分割が可能となっている。

以上述べた構成により，実現すべき機能仕様を明確に状態

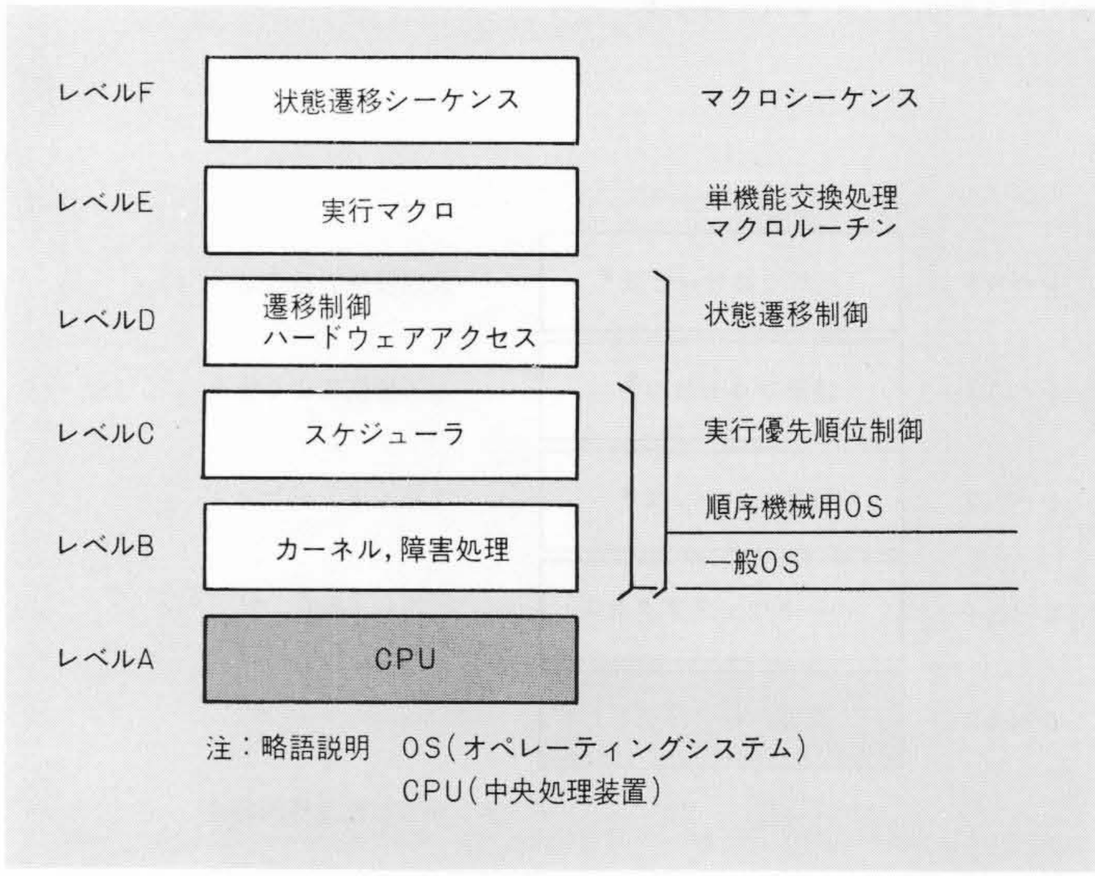

図6 実現手段の階層分割 実現手段（プログラム）を統一的手法により，CPUハードウェアを中心に5階層分割できる。

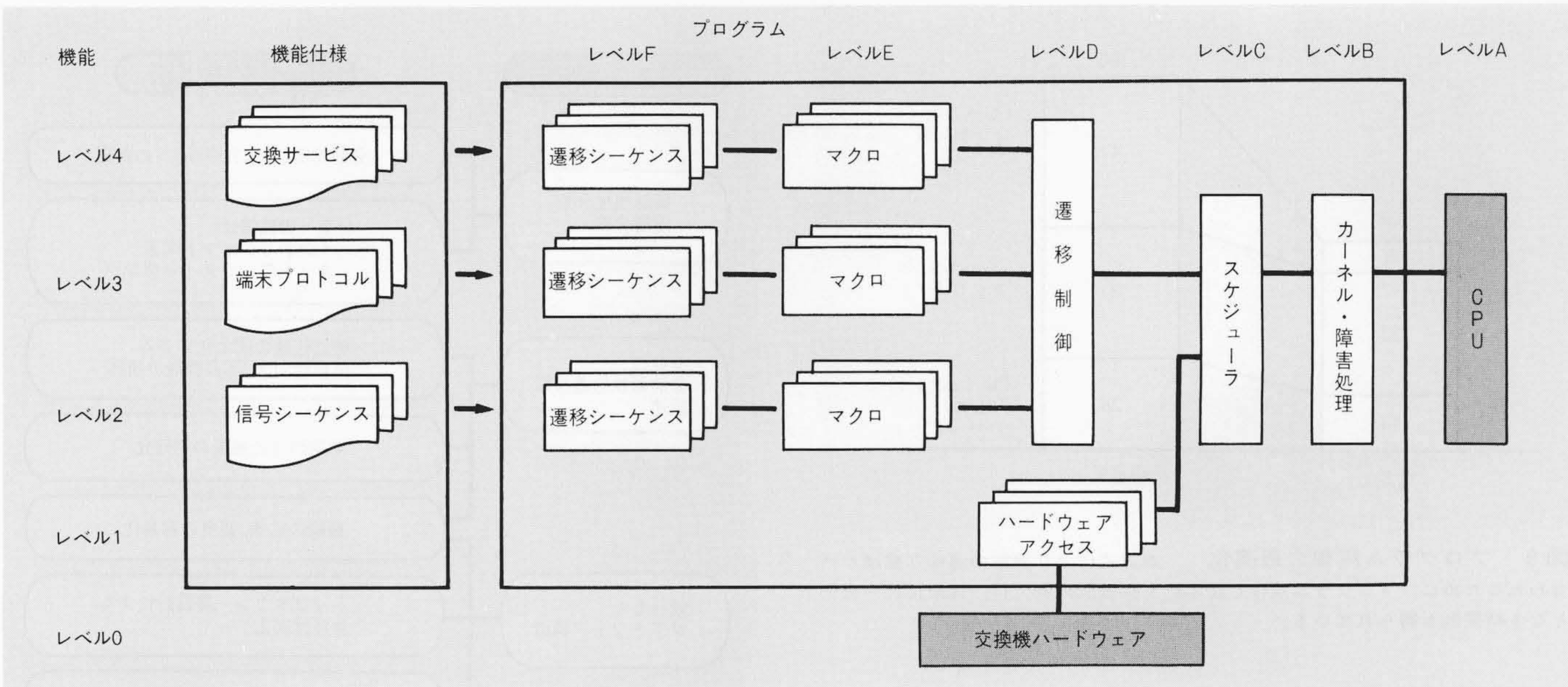

図7　2次元階層化ソフトウェア構造　交換機能のレベル0～4の階層化と，実現手段のプログラムのレベルBからFまでの階層化を直交させた統一した2次元階層化ソフトウェア構造となっている。

遷移図で記述し，それに基づいたトップダウンなソフトウェア設計を可能とし，所期の目標を達成している。

5　本方式の効果

(1)　多様な機能の実現の容易化と将来への拡張容易性

本方式では仮想交換機の概念の導入による多階層化と，端末種別ごとに順序機械制御を独立化しており，それらの複数の順序機械の組合せにより交換機能の実現を図っている。その結果，状態の数は組合せ方式により従来に比べ半減しており，かつ状態間の遷移処理内容も階層分割化により半分以下となっている。

今後出現する多様な端末，交換スイッチのネットワーク化などの動向に対して，独立した順序機械を付加，あるいは取替えにより，個別にかつ少ない状態遷移数で対応を可能としている。これにより，将来に対する高い拡張性をもっている。

(2)　用途別交換ソフトウェアの共通化促進

状態遷移図により，交換機能の図示化が図られていることにより，用途別に異なる機能部分と共通部分とが明らかにでき，その結果，同一機能部分のプログラムの共通化が図れ，ソフトウェア財産の有効活用が促進される。

(3)　機能の拡張，変更の容易化

機能の変更，追加のニーズは，顧客ごと，用途ごとに多く生じている。それに対し本方式では，該当する機能部分を状態遷移図で明確化できるとともに，対応するプログラムも明確に関連づけられる。

これにより，高い機能拡張，変更の容易性が得られている。また，既存モジュールの組合せにより，多様なサービスの実現も可能である。

(4)　分散制御による規模拡張性への対応

図8に示すように，小規模から中，大規模に対応して複数のマイクロコンピュータによる分散制御方式の導入により，高い規模拡張性が得られるが，ソフトウェアはそれに応じて階層レベルでの分割，仮想機械での分割により対処が可能である。また，コンパイラの導入により，規模によるデータ長の変化にも容易に対応が可能となっている。

(5)　プログラム規模の減少化

従来手法で開発された交換プログラムの規模と本方式によるプログラムとを比較すると，**図9**に示すように約半減している結果例が得られた。

この削減の効果は，第一には，状態遷移図の複雑さが大幅に軽減されたために，それに対応するシーケンステーブルが大幅に簡単化されたことにある。特に，仮想交換機のレベル4では顕著で，全体の約4割を占めていたテーブル規模が約$\frac{1}{5}$になっている。第二には，入出力部のレベル2，3に状態遷移制御方式を導入した結果，レベル1を含めた規模が半減している。更に，交換機能を状態遷移図ですべて統一して図示化した結果，プログラム間の機能の重複が避けられていることも減少した要因の一つである。

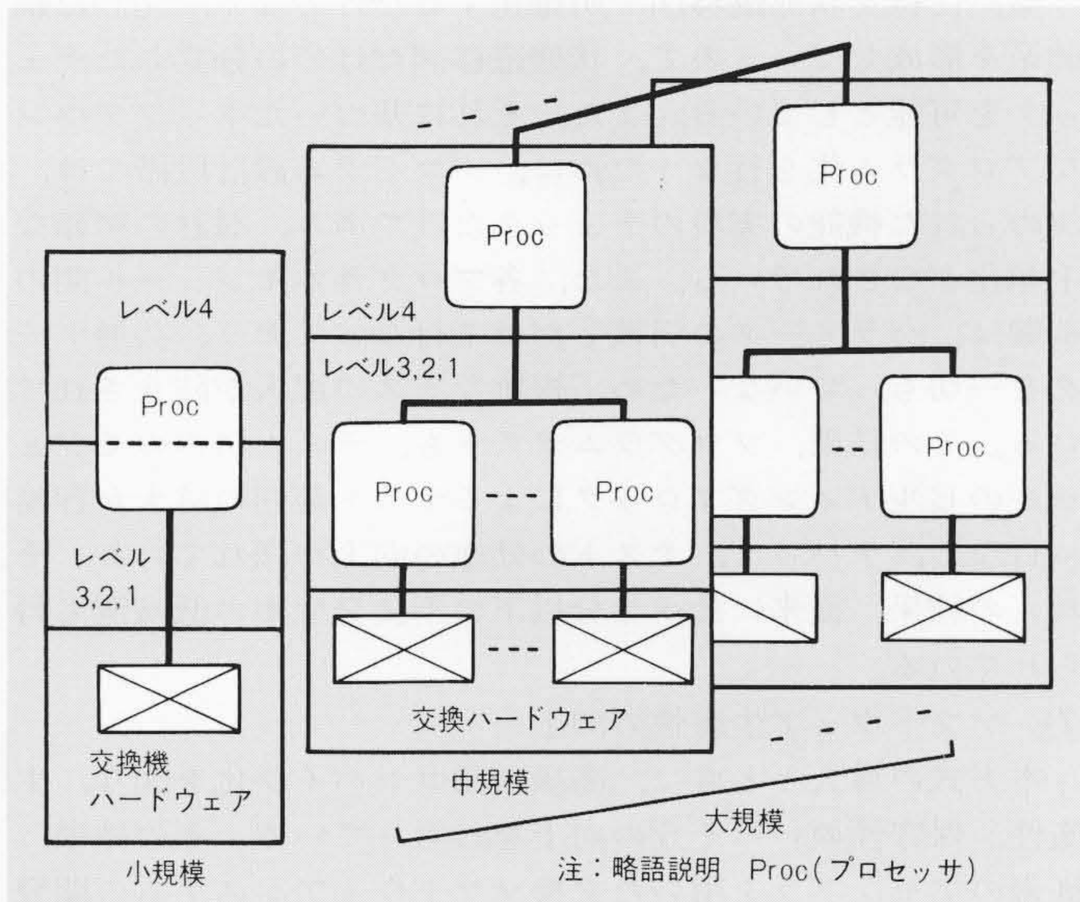

図8　分散制御方式との対応　小規模から中，大規模に対応して複数のプロセッサの分散制御により構成されるが，それに対応して階層分割，仮想機械分割を容易に行なえる。

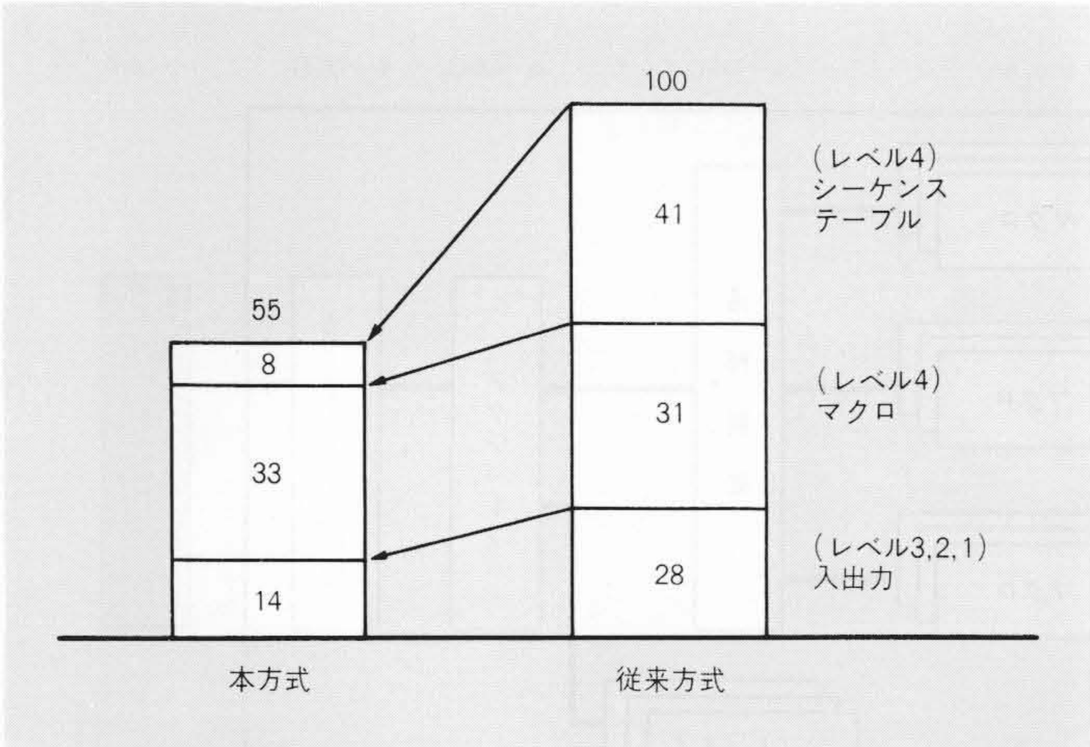

図9 プログラム規模の最適化 本方式により機能の適切な整理が行なわれるために，プログラム規模が基本的交換機能の例では，従来に比べ55%となる結果例も得られている。

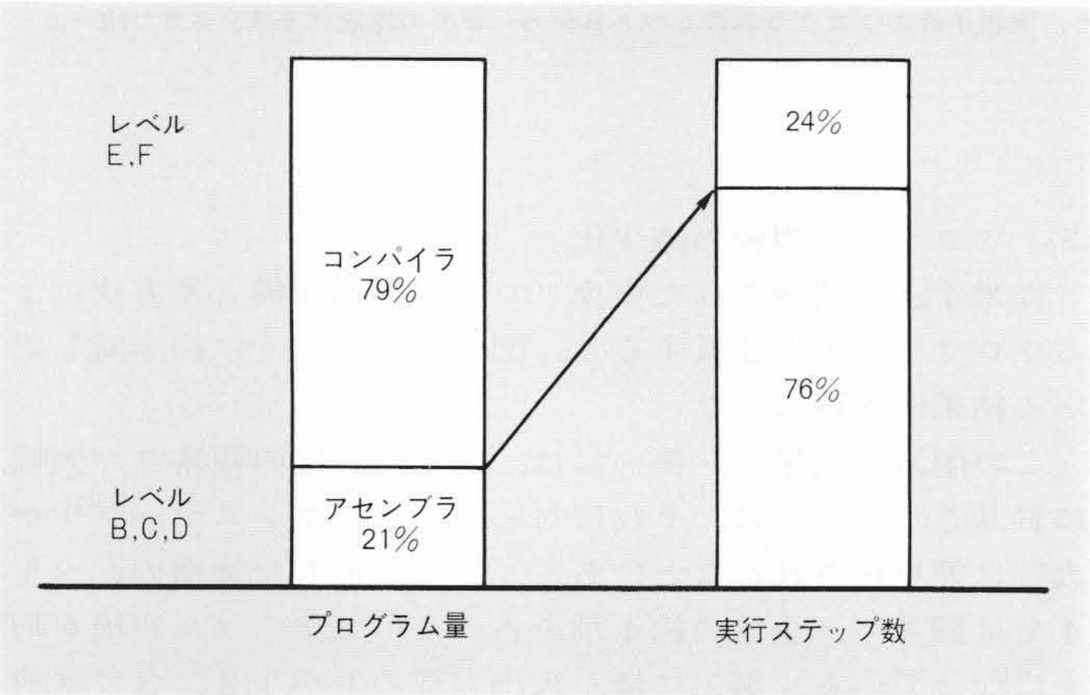

図10 コンパイラの導入と性能への影響 実行頻度の高いレベルB，C，Dをアセンブラとし，プログラム量の多いレベルE，Fをコンパイラとすることにより，生産性と性能のバランスを図っている。

(6) ソフトウェアの信頼性の向上

機能仕様を状態遷移図で明確化するだけでなく，閉じた制御系を形成しているので，状態遷移図だけでの独立したチェックを可能としている。また，それに基づいたトップダウンなプログラム化を行なうために，プログラム設計段階では，決められた機能の実現のチェックだけで済み，設計の明確な手順化がなされている。また，各プログラムモジュール間の制御は，パラメータの引渡しだけで行なっており，共通データを一切もっていないため，複雑なミスの混入が防止されている。その結果，プログラムテストも，テスト済みのモジュールのビルディングブロックによるテスト範囲の拡大が容易に行なえ，デバッグ，テストの効率の向上が図れている。それらの結果，従来に比べ半分以下の不良発生率の低減例も得られている。

(7) ソフトウェア生産性の向上

本方式の導入とともに，高級言語コンパイラ化を図り，生産性，保守性のいっそうの向上をねらっている。その結果，従来のアセンブラを用いた交換ソフトウェアシステムの開発に要した工数と比べると，同一機能部分で約3倍以上の生産性向上の結果例が得られている。この効果のうち，本方式による部分は約2倍程度と考えられる。この結果は，前述のような設計手順の段階化，プログラムモジュールの高い独立性，更に設計品質の向上などによる効果である。

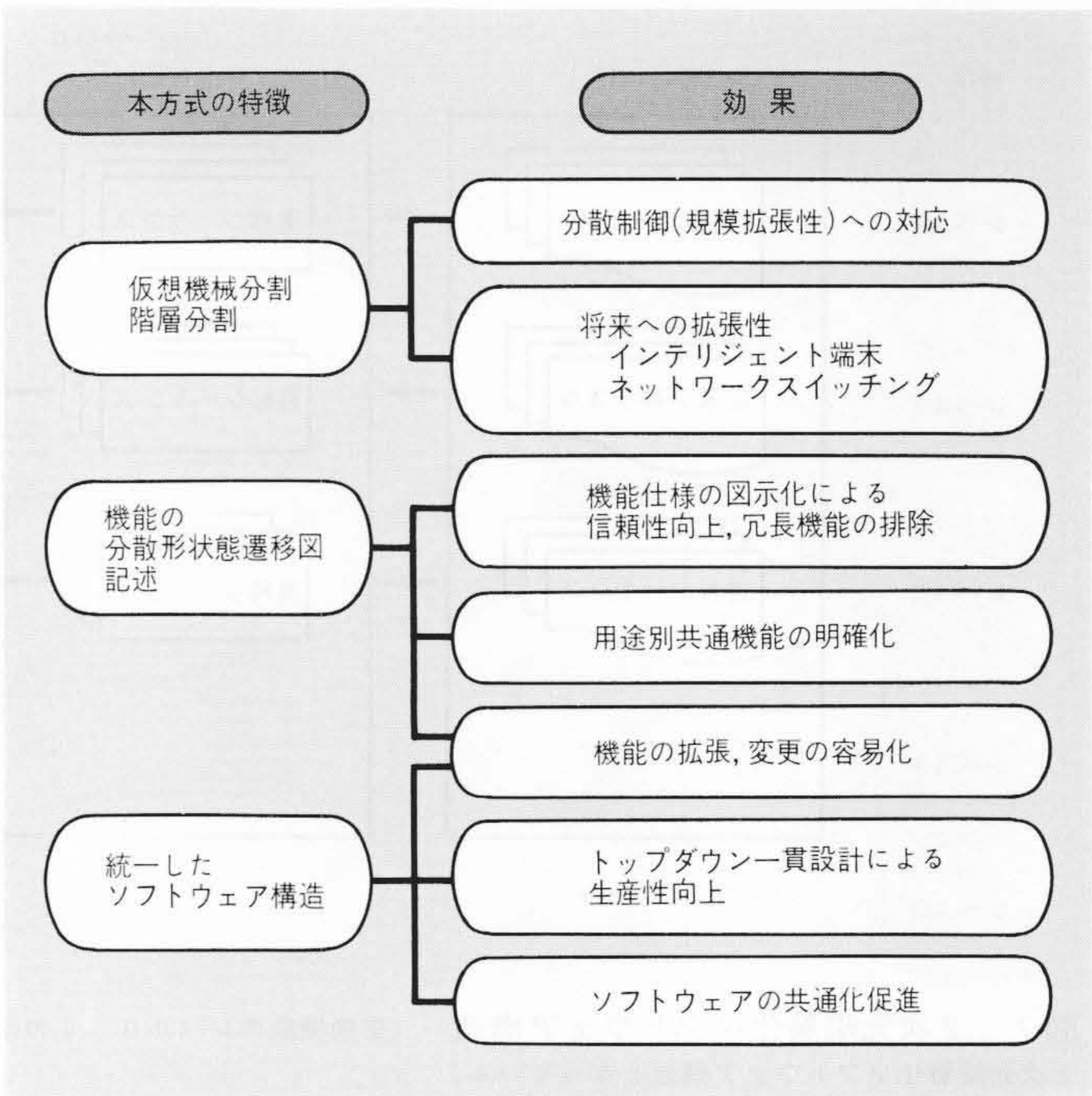

図11 本ソフトウェア方式による効果 本ソフトウェア方式により，高い拡張性，信頼性とともに生産性向上が得られる。

(8) 性 能

高級言語の導入とともに，従来のアセンブラ言語に比べ性能低下のおそれがあるが，それを防ぐために使用頻度の高い一部のプログラムのアセンブラ言語化を行なっている。

本方式により，その部分はレベルB，C，D内に局所化でき，明解な構成化が図れている。**図10**に示すように，部分的なアセンブラ化により，性能上は従来に比べなんら劣らないものにできている。

以上述べた関係を**図11**に示す。

6 結 言

本方式により，大規模かつ複雑化する交換ソフトウェアに対し，統一的な構造，設計手法を導入することができた。その結果，従来に比べて大幅に生産性，信頼性の向上が得られるとともに，機能の変更，拡張も格段に容易となり，多様なサービスへの対応，更に今後の交換システムの発展に対応できるものとなすことができた。

本方式はPABX(私設交換機)だけでなく局用交換機にも適用が可能で，その両者の統合した開発を可能とすることができる。

参考文献

1) 佐藤，外：D10形電子交換機の方式概観，日立評論，**54**，11，985～990（昭47-11）

2) CCITT："Recommendation Z. 101"，Orange Book，Ⅳ，4（1977）
状態遷移図記法のCCITTの勧告を述べている。

3) 桧山，外：電子交換機の分散型状態遷移方式の提案，電子通信学会交換研究会，SE78-118，(1979)
本方式の基本構想を示している。

4) 桧山，外：分散型状態遷移方式による交換ソフトウェアの一方式，電子通信学会交換研究会，SE81-85，(1981)
本方式の具体的方法，結果について示している。