情　報　シ　ス　テ　ム 49－3
（１９９４．　５．　２０）

層別化および可視化による情報処理システムの構築

富沢研三（三菱電機），出口博章（三菱電機ビジネスシステム）
魚田勝臣（専修大学），田村幸子（九州産業大学）

ビジネス系のシステム構築において、システムを要求する側（エンドユーザ）の立場に立ち、かつシステム開発部門におだやかな進歩を求める方策について述べる。従来から、ソフトウェア開発の生産性や品質の向上に対して各種方法論や開発環境の提案がされている。しかし新しい手法が現場部門になじみ難く十分効果が発揮されていないのが実情である。

そこで、企業活動を業務特性に基づいて層別化し、それぞれに合致した開発方法を選択することを考えた。また、企業の顔となる特徴的なシステムについてはエンドユーザの参画が不可欠であるとの認識から、従来とは違った方法でソフトウェアを可視化することを考えた。

SYSTEM DEVELOPMENT METHOD BASED ON CLASSIFICATION AND VISUALIZATION

Kenzo Tomizawa (MITSUBISHI ELECTRIC CORP.)
Hiroaki Deguchi (MITSUBISHI ELECTRIC BUSINESS SYSTEM CORP.)
Katsuomi Uota (SENSYU UNIV.)
Sachiko Tamura (KYUSYU SANGYO UNIV.)

This paper proposes the new method for business systems development. The method is based on the classification of corporate business activities and the uniquevisualization of application programs. Our method first classifies corporate activities, and then defines the most appropriate development method for each of classes. By this approach, it is shown that about 60% of the system can be replaced by the package software, hence the volume of application development can be reduced to 40%. The method also allows end users to easily participate in system development. This is achieved by vbisualizing application programs. Our visualization differs from the previous ones in that they are based on the diagrams and the procedures.

1　まえがき

企業のリストラクチュアリングの影響を受けて、情報システムも変革を求められている。また、ハードウェアのダウンサイジングの進行とともに情報処理システムにおいてソフトウェアが占める割合はますます増大している。

ところが、ソフトウェア開発の生産性・品質に飛躍的な向上は見られない。いろいろな方法論や開発環境が提案されているものの、十分効果が発揮されていないのが実情である。その一つの原因に、新しい開発手法が現場部門になじみにくいことがあげられる。

われわれは情報システムを要求する側（エンドユーザ）の立場に立ち、かつ情報システム部門におだやかな進歩を求める方策を研究している。このため、企業活動を業務特性に基づいて層別化し、それに合致した開発方法を選択することを考えた。また、企業の特徴を出し、長期にわたって改良・維持される業務については、ユーザの深い参画が不可欠であるとの認識から、従来とは違ったやり方でソフトウェアを可視化することを考えた。

2　システム開発の現状

2．1　情報化投資の増大と情報システムの重要性の認識

企業の経営環境が目まぐるしく変わる中で、情報システムは拡大・多様化・複雑化し情報化投資は増加の一途をたどってきた。ソフトウェアの比重の増大の要因をまとめると次のようになる。

(1)情報システムに対する投資の拡大

企業はこれまで、事務処理の省力化・迅速化といった合理化目的に情報処理システムを構築してきた。近年になってこれらの主なものは実現され、企業はその目的をより高度な戦略目的に重点を移しつつある（図1）。戦略目的を遂行する情報処理システムは合理化目的に較べて、複雑で大規模となり投資額も膨らんでくる。

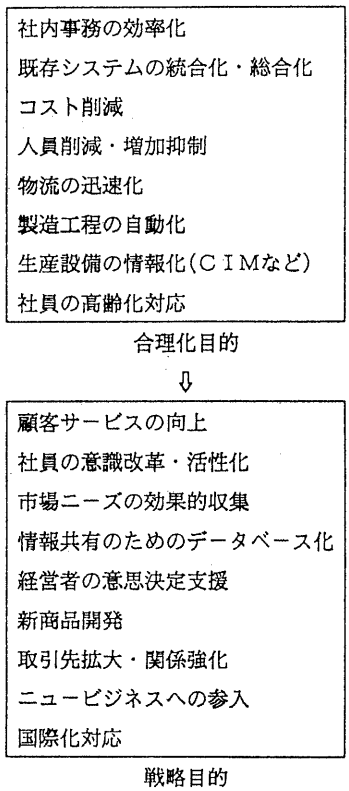

図1　合理化目的から戦略目的への重点の移行

(2)ソフトウェアに対する投資の増大

情報処理システムの構成要素であるハードウェアや基本ソフトウェア(オペレーティングシステム)は、標準化・均一化の傾向にあり、機器による機能・性能の差はなくなってきている。これに対してアプリケーションソフトウェアやサービスがシステムの優劣を決めるキーになりつつある。他社を差別化できるシステムを構築することが、企業の存続のために必要となり、その結果ソフトウェアに対する投資は今まで以上に増大していくと考えられる(図2)。

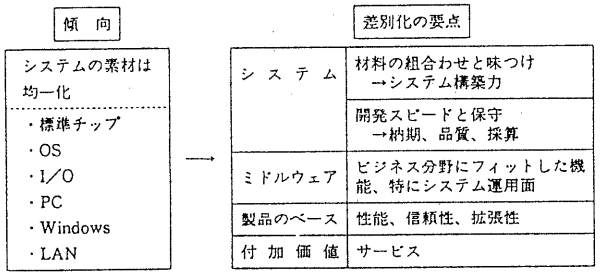

図２　システムの差別化の要点

２．２　システム開発技術の現状

現在採用されている情報処理システムの実現手段には、以下の三つの種類があり、それぞれ特徴を持っている。

(1)個別開発

それぞれの企業の要求に合わせてシステムを設計し、ソフトウェアを手作りする形態である。オーダメイドとかカスタムメイドと呼ばれ、自社開発と外注する場合がある。

一般にライフサイクルモデルとしては、開発作業を幾つかの工程に分割し、各工程の中間生産物を伝承するウォータフォールモデル（図３）が適用されている。

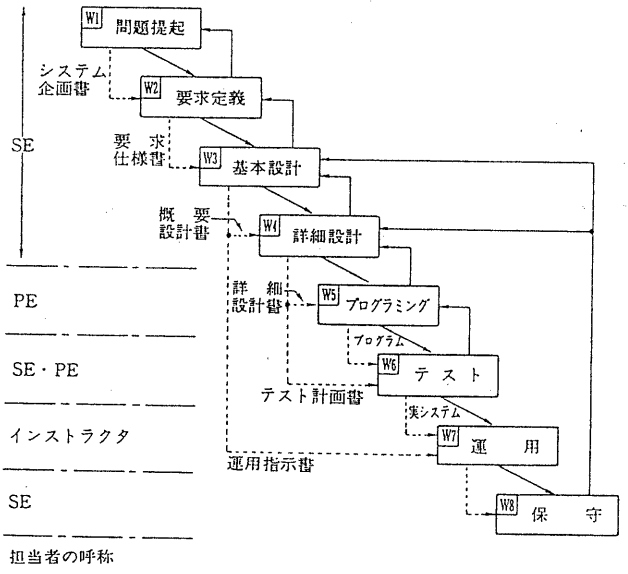

図３　ソフトウェアライフサイクルのウォータフォールモデル(WFM)

(2)パッケージソフトウェアの利用

高品質のソフトウェアを低コスト・短期間で得る手段としては、パッケージソフトウェア（パッケージ）の適用が有効である。しかしパッケージは機能が固定されているので、自社の要求に適合させるための改造作業（カストマイズ）を行わねばならぬことが多い。したがって、パッケージソフトウェアはルールが比較的安定している業務のシステム構築に適している。

(3)ＥＵＣの適用

ＥＵＣは現場の業務を熟知しているエンドユーザが自らシステムを構築することができるため自分の希望するシステムを実現できる。具体的には、コンピュータに蓄えられている基幹業務のデータを、第４世代言語（４ＧＬ）や簡易言語などを使って構築することである。

しかし、ＥＵＣでは大量のトランザクションを効率よく高品質で処理するといった大規模なシステムの構築には向いていない。

３　経営活動の層別化モデルと情報システム実現手段の選択

情報処理システムを効率よく実現する場合、対象とする業務を情報処理システムの観点から分析し、その特性に適合する実現手段を選択する必要がある。

情報処理システムには、経営や業務の改革・改善にともなって激しく変化する部分と変化がゆるやかな部分とがある。たとえば、営業システムのように企業戦略の変化に応じて情報処理システムも改革・改善する必要がある領域では、システムの保守や改良が重視される。一方、経理、給与システムのように業務ルールが比較的安定している領域ではシステムは長期にわたり継続して使用される。このような業務の特性に応じたシステムの実現手段を適用することにより、現場の要求に合致したシステムを効率よく実現し、変化に対応することが出来ると考える。

3．1　企業活動のモデル化および層別化

図4に示すように企業活動をラインとスタッフ業務に分け、後者をさらに業務管理業務と経営支援業務とに分類して考える。

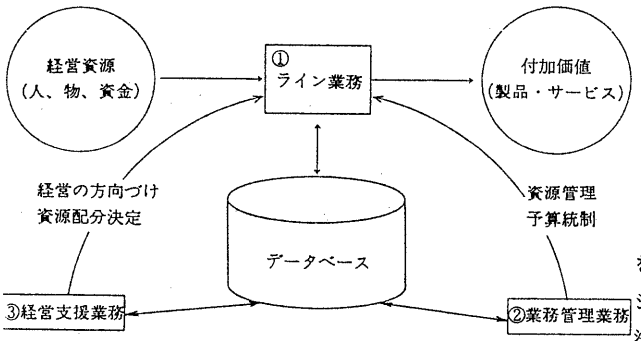

図4　企業活動のモデル

(1)ライン業務

ライン業務は企業の目的である利益の源泉を産出する業務である。一般に営業、仕入、生産、物流などの部門が担当する。伝票を中心としたトランザクションの量が多く、処理速度と品質が要求される。企業の中心となるデータベースは、このライン業務で生成、更新、削除される。

(2)業務管理業務

業務管理業務はライン業務での経営活動を管理する業務である。一般に財務管理、設備管理、資材管理などを行う、経理、総務などが主たる担当部門である。ライン業務が作り出すデータベースをインプットとして、企業の資産の動向を計数処理し報告するシステムである。一般に、業務のルールは、異なる企業間でも同一でしかも定期的に繰り返し実施される。

(3)経営支援業務

経営支援業務は各種のデータを分析し企業経営の方向づけを支援する業務で企画・計画部門が担当する。ライン業務、業務管理業務が作り出す社内のデータベースや外部のマーケットのデータベースなどをインプットとしてデータベースを検索・加工し、シュミレーションなどを行い、調査・分析報告書を作成する。報告書の種類や形式は多様であり、試行錯誤的なシステムである。

3．2　業務の特性に応じた実現方法の選択

情報システムを構築する場合、システムの構築方法を一元的に考えるのではなく、企業の業務を情報処理システムの観点から層別化しその特性に応じた構築方法を採用する方が効率的である。そこで、3．1で検討した企業活動の層別化と2．2でのべたシステムの実現方法を図5のように組み合わせることを考えた。

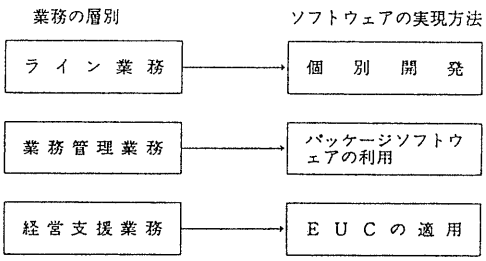

図5　企業活動の層別化とｿﾌﾄｳｪｱ実現方法の対応

情報処理システムの実現方法の選択は、これまで暗黙のうちに行われていた。しかし、業務の層別化を通じて整理していなかったため、担当のＳＥが思いつくままに選択していた。例えばライン業務に対してパッケージを適用することを考えたり、逆に業務管理業務を個別に開発していた。表1のように明示的に分割した選択法を適用することによって、効率的なシステム開発が行える。

表1　業務領域とソフトウェアの実現方法の選択

業務領域	領域の特徴	ソフトウェアの実現方法
ライン業務領域	ライン業務は企業ごとの戦略・戦術を含んでいる。情報処理システムは企業ごとに異なり特徴を持ったものでなければならない。また、業務の変革とともに進化しなければならない	個別開発による構築：ソフトウェアの可視化による構築法と開発環境を適用し、エンドユーザの参画を得て構築
業務管理領域	この種の業務は基本的に企業間で共通しているルールに則って処理するものであり、情報処理システムとしては一番安定したシステムである	パッケージをカストマイズして実現：今後さらに機能が成熟し完成度の高いものが出てくると考えられる
経営支援領域	この領域の業務は報告書の表現形式が多様であり、直感的に理解できるグラフや表が求められる。データベースに対する多角的な検索が必要	ＥＵＣによる実現：ＯＡソフトや４ＧＬを組み合わせた積み木型構築

4　可視化技法

ここでは層別化されたソフトウェアのうち、広い意味でのプログラミング言語を使ってプログラムを個別開発するライン業務の生産を合理化する方法について考察する。合理化の要点はプログラムをエンドユーザにも理解できる記述形式にすることである。

4．1　可視化技法の必要性

ライン業務に対する情報処理システムは、その性格から使い易く性能の良いプログラムを作る必要があり、ユーザの要求を細部にわたって正確に反映したものでなければならない。このようなプログラムを開発するのに４ＧＬやパッケージのような実現手段を用いると機能・性能・操作性の面で問題が発生する可能性が大きい。したがって、このようなプログラムはプログラミング言語を用いて個別に作成した方が、エンドユーザの満足度は大きくなる。

しかし、システムの開発はコンピュータの専門家によって行われ、どんなシステムが出来上がってくるかはエンドユーザの目でチェックできない。

システムの内容がエンドユーザや他のシステム開発者にとってブラックボックスになっているのは、次のような問題を引き起こす原因となり、生産性を悪化させている。

(1)仕様のずれによる手戻り作業の発生

システムの開発はエンドユーザ、ＳＥ、プログラマと複数の人手と工程を経て完成される。各工程間は仕様書、設計書等のドキュメント（文書またはリポジトリ）で情報が伝達される。しかし、業務の内容や操作方法を正確にドキュメントに反映することは難しく、細部の決定はＳＥやプログラマに任せることが多い。このためエンドユーザが望んだとおりのプログラムが出来上げってこない可能性があり、プログラムの改修作業が発生する。

(2)設計変更、システム変更時の非能率さ

企業を支えるラインシステムでは、外部環境の変化に起因する設計変更やシステム機能拡張等の変更に迅速にプログラムを修正して対処することが重要となる。しかし、プログラムの修正はプログラムが可視化されていないためその内容に熟知した者（一般にはプログラム開発に携わった特定の人間）に頼らざるを得なくなるのが実情である。

(3)プログラム流用の困難さ

プログラム開発者側から考えると、一度作成したプログラムを別な類似システムの開発時にも流用できれば生産性は非常に向上する。しかし、プログラムが可視化されていないため個人レベルあるいは開発チームレベルといった狭い範囲でしか行われていないのが実情である。

４．２　プログラムの構造分析

従来プログラムの可視化について、ファイルレイアウトや画面レイアウトなどの部品レベルの作成ツールや手続き部の日本語表現など局部的な試みは種々行われている。

ビジネス系のプログラムはデータの集計、分類といった比較的単純な処理を行うものが多く可視化が比較的容易と考えられる。そこで、プログラムの記述要素を既存のプログラム言語にとらわれることなく、記述内容の果たす役割の面から分類すると、図６に示すような５種類の定義部分に大別することができる。

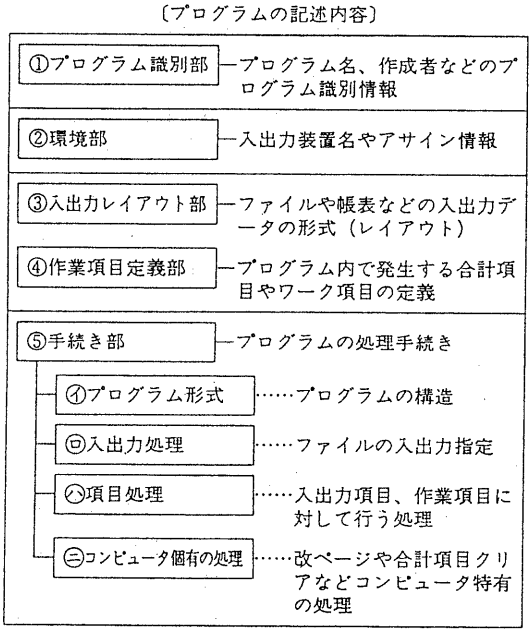

図６　プログラムの記述内容

図６に示したプログラムの記述内容のうち①～④についてはプログラム名や項目名などの定義であるから機能的には難解なものではない。

問題は手続き部である。手続き部ではａ～ｄにそれぞれ内容の異なる処理が混在しており、１ステップずつ解読していかないと処理の内容は分らない。特に、プログラム形式(ａ)とコンピュータ特有の処理(ｄ)はコンピュータの知識がないエンドユーザにとっては非常に難解である。

４．３　可視化プログラムの構造

図６に示した現状のプログラムの構造分析を通じて、プログラムの記述の大半は項目に関する定義や処理の指定であり、可視化によって基本的にエンドユーザにも理解できるものであることが分った。可視化の要点としては、以下のようなことがあげられる。

(1)プログラム内での記述項目を減らす

プログラムを分り易く表現するための基本的な手段はプログラム内での記述量を極力減らすことである。そのためには、システム内で共通部品として扱える項目に関する定義や、項目の組み合わせで構成されている入出力レイアウト部をプログラムから独立した外部定義とする。また、プログラムの構造についても画面処理プログラムや帳票処理プログラムといったようにある程度パターン化することにより構造に関する記述は不要となる。

(2)項目の部品化と項目属性定義

一つのシステムは多数のプログラムから構成され、個々のプログラムでは共通の項目（データ項目）を処理する場合が多い。こうした共通項目は、ライン業務の骨格を成す重要な項目であるから、これらを部品扱いとした方がシステムの開発、保守の面で有利である。

項目の部品化を、ここでは項目の属性定義と呼ぶ。項目の属性定義はカプセル化と呼ばれる手法で項目の名前や意味、あるいは計算式や値など項目の処理を含めて定義しておく。このことによりその項目を扱う全てのプログラムは項目を選択するだけで、項目に関する処理の指定はプログラム内で記述しなくて済む。項目の属性定義において重要なことはシステム中の全ての項目について定義するのではなく、ラインシステムの共通部品的な項目に限定することである。

(3)入出力レイアウト部の図式表現

ファイルレイアウトや画面レイアウトといった入出力レイアウト部については、レイアウトのイメージが一目で理解できる図式表現とする。これは従来から広く行われており技術的に問題ない。しかし、レイアウト作成時に項目の属性情報を参照したり、作成されたレイアウトは項目の属性定義と関連をもたせることが重要である。

(4)コンピュータ特有処理の分離と指定

改ページや合計項目のクリアなど直接業務と関係はないがプログラム内では指定の必要なコンピュータ特有処理については、プログラムの骨格から外し分離させる。このことにより本来業務で必要な項目に関する処理とコンピュータ固有処理が混在してプログラムが難解となっていた問題を解決することができる。コンピュータ特有の処理は、基本的にコンピュータの専門家だけが指定できれば良いと考える。

以上(1)~(4)で考察したことからプログラムの構造を可視化したのが図7である。この図のうち、②および③は外部定義し、プログラムから分離する。また、④のプログラム形式はあらかじめ用意してあるプログラムパターンから選択することにより、プログラム内での記述は不要とする。このことによりプログラムの中の記述はプログラム形式の各ブロックの詳細処理に関する記述だけとなる。

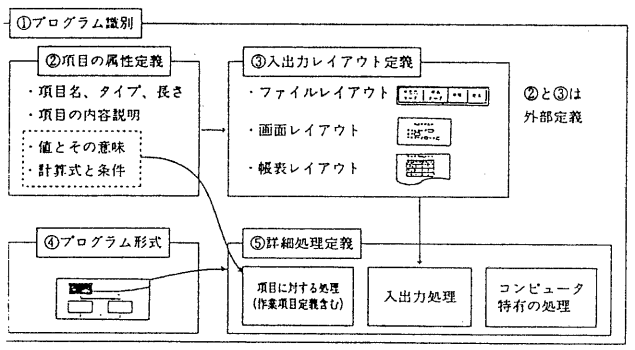

図7　可視化プログラムの構造

4．4　可視化を実現するシステムの構成

可視化プログラムを作成するための開発環境のシステム構成を図8に示す。

(1)情報定義部

情報定義部は、項目の属性定義、入出力レイアウト定義、プログラムパターン定義から構成される。

入出力レイアウト定義は項目の属性定義を参照しながら作成する。情報定義部で作成された属性定義情報、レイアウト情報、プログラムパターン情報はシステム内のプログラム部品として扱われ、次のプログラム作成部で参照される。

(2)プログラム作成部

プログラム作成部は、情報定義部で作成されたプログラム部品を参照しながらプログラムを組み立てていく。その手順を以下に示す。

①プログラム識別情報を指定する。

②プログラムで扱う入出力情報をレイアウト情報の中から選択する。

③プログラムの構造を決定するために、そのプログラムに類似した処理パターンを、プログラムパターン情報から選択する。

④プログラムパターン内の各ブロックに対して①で選択したレイアウト情報を対応づけ、ブロックでの記述内容を規定する。

⑤各ブロック毎にレイアウト情報内の項目を選択し、項目を処理する順番を指定する。また、項目の処理について項目の属性情報で定義された処理内容に変更があれば変更指定を行う。更に、プログラム内で発生する項目があれば、その定義とその処理内容を指定する。

(3)ソース生成部

ソース生成部は(2)で作成したプログラム情報をもとにプログラム言語対応のソースプログラムに変換する。ソース生成部で直接オブジェクトプログラムを作成する方式を採用することもできる。

ソースプログラム内での項目名はプログラム情報で指定された日本語項目名からマッチングをとり自動的に命名する。

5　あとがき

企業の業務を三つの階層に分類して、それぞれの性質に適合したソフトウェアの実現手段を選択する方法と、個別開発するソフトウェアについて可視化によってエンドユーザに理解できる表現法を提案した。可視化については開発環境を明らかにし実現可能であることを示した。

この方法論と開発環境を適用することを通じて、エンドユーザがシステムの構築法を学び、エンドユーザデザイン（ＥＵＤ）に道を拓く。

層別化の方法はすでにＳＩによって適用されつつある。可視化のシステムは現在構築中である。完成すれば、諸元や効果などについて改めて報告する。

参考文献

(1) 魚田勝臣："ソフトウェアリユースモデルと開発環境"，専修経営学論集，Vol.57, 1993, pp.88-117.

(2) 通商産業省編："特定サービス産業実態調査報告書　情報サービス産業編" 昭和63年，平成元年および平成２年

(3) 浦　昭二編："情報システムの教育体系の確立に関する総合的研究" 平成３－４年度科学研究費補助金（総合研究Ａ）研究成果報告書，慶応義塾大学，1992，p.17.

本文から引用していないが次の文献を参考にした。

(4) 田中克己："優位に立つSIベンダー"，日経コンピュータ，1991.5.20，pp.76-92.

(5) 中原啓一，加藤栄護編："システムズエンジニアハンドブック"，オーム社，1991, p.120.

(6) 岩尾達男："ダウンサイジングとアウトソーシング"，日本生産性本部，1993, pp.219-221.

(7) 原田実監修："CASEのすべて"，オーム社，1991, pp.3-5.

(8) 魚田勝臣："オフィス情報システムのグラフ記述の一方式"，情報処理学会論文誌，Vol.27，No.2, 1984, pp.235-242.

(9) Dologite, D. G.: "Using Computers", Prentice-Hall, 1992, p.333.

(10) 魚田勝臣他："インライン処理指向のプログラミング言語"，情報処理学会論文誌，Vol.21, No.6, 1980, pp.433-441.

(11) 富沢研三他：SIにおける中規模システム開発手法について"，第46回情報処理学会全国大会講演論文集，1993.3, pp.5-339-340.

(12) 白石剛士他："リサイクルを基盤としたシステム開発，保守環境" 第46回情報処理学会全国大会講演論文集，1993. 3, pp.5-341-342.

(13) 魚田勝臣他："ユーザニーズに直結した仕様書のありかたについて"，第46回情報処理学会全国大会講演論文集，1993. 3, pp.5-343-344.

(14) Kimura, T. D.: "Hyperflow: A Uniform Visual Language for Different Levels of Programming", The Proceedings of ACM Computer Science Conference, Indianapolis, Feb. 1993.

(15) Hils, D. D.: "Visual Languages and Computing Survey: Data Flow Visual Programming Languages", Journal of Visual Languages and Computing 3: 1, 1992, pp.69-101.

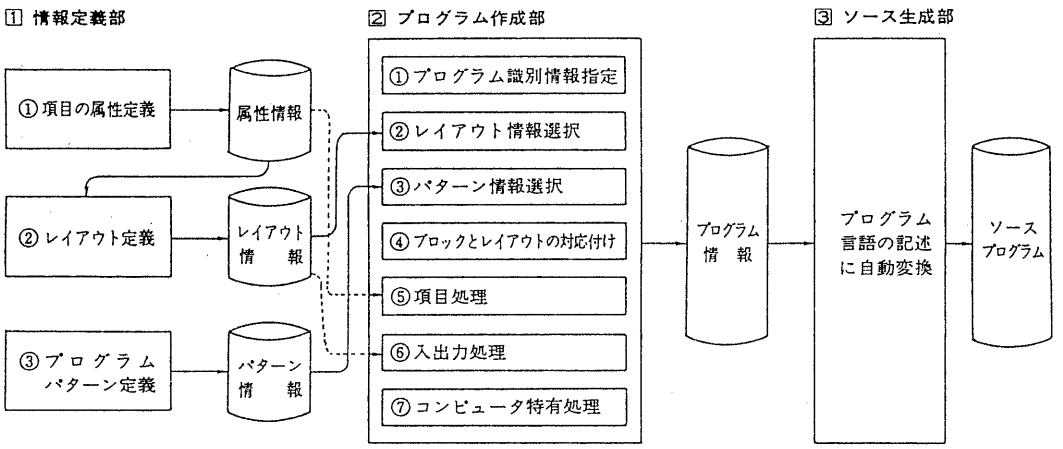

図８　ビジブル化技法のシステム構成