

日立製作所のアプリケーション開発支援体系 “CAPSDF”

—“CAPSDF”を構成する技法・方法論・ツールの概観—

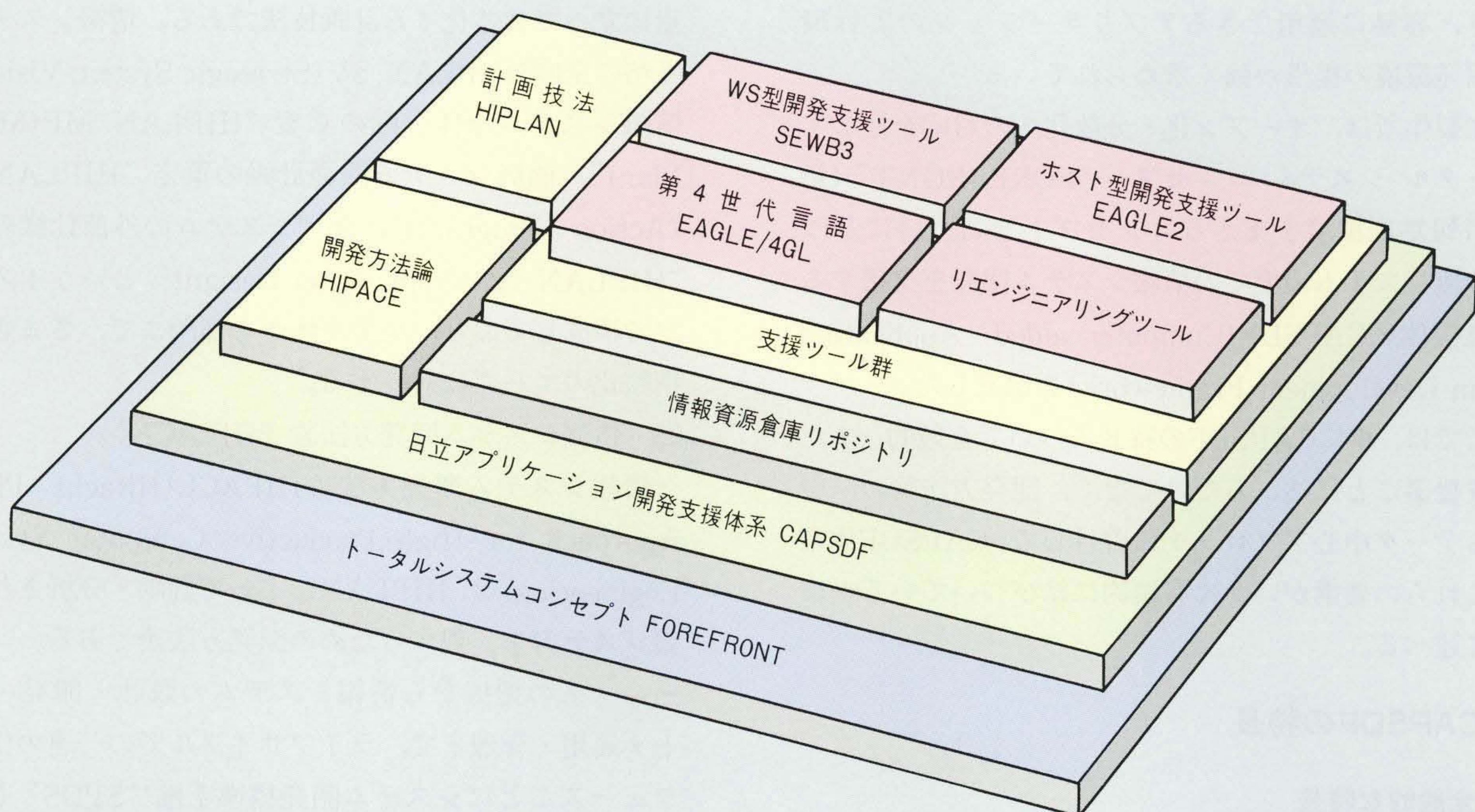
Concept of Hitachi Application Development Framework

西尾高典* *Takanori Nishio*

中原俊政* *Toshimasa Nakahara*

今城哲二* *Tetsuji Imajo*

津田道夫** *Michio Tsuda*



注：略語説明

HIPLAN (Hitachi Integrated Planning Procedure for Information System), HIPACE (Hitachi Phased Approach for High Productive Computer System's Engineering), SEWB3 (Software Engineering Workbench 3), EAGLE2 (Effective Approach to Achieving High Level Software Productivity 2), EAGLE/4GL (EAGLE/4th Generation Language), CAPSDF (Computer-aided Application System Development Framework)

日立製作所のアプリケーション開発支援体系CAPSDFの概観
アプリケーション開発を強力にバックアップする。

CAPSDFは、技法・方法論・支援ツール・リポジトリが一体となってア

CAPSDF (Computer-aided Application System Development Framework) は、トータル システム コンセプト“FOREFRONT”に基づく分散システム環境での情報システム開発を支援する開発支援体系である。技法・方法論・支援ツール・リポジトリといった開発支援に欠かせない要素技術を統合し、これらが一体となって開発を支援する。クライアントサーバ型の開発環境であること、大型コンピュータ中心の大規模システムからワークステーション業務

の開発までマルチターゲットであること、業界標準のトースタモデルに準拠していることなどの特徴を持つ。また、方法論の一つとしてデータ中心アプローチを採用しており、支援ツールやリポジトリが、この考え方に基づく開発作業を強力に支援している。このほか、プロトタイピング技法や構造化技法、部品再利用技法、プロジェクト管理技法なども支援している。

* 日立製作所 ソフトウェア開発本部

** 日立製作所 ビジネスシステム開発センタ

1 はじめに

情報システムは、企業戦略のキーとしてますます重要さを増す一方、情報システムへの要求も多様化し、それらの要求に対応するために、クライアントサーバシステム、オンライン トランザクション プロセッシングなど、コンピュータ環境が著しく進化しつつある。このような状況の中で、開発言語やシステム規模に応じて柔軟に構築でき、容易に運用できるアプリケーションの実行環境・開発環境の提供が強く求められている^{1),2)}。

日立製作所は、オープン化・分散化への対応を柱とするトータル システム コンセプト“FOREFRONT”(新しい情報処理環境を支えるコンセプトを言う。)に基づき、分散システム環境での情報システム開発を支援する開発支援体系CAPSDF(Computer-aided Application System Development Framework)を確立した。

ここでは、単にCAPSDFの特長について全般的、また各構成要素ごとに述べるだけでなく、開発方法論の一つであるデータ中心アプローチに着目して、CAPSDF体系上でこれらの要素がいかに有機的に結びついているかについて述べる。

2 CAPSDFの特長

2.1 全般的な特長

(1) 情報システム開発の体系化

CAPSDFは、情報システムの計画・分析から設計・開発、運用・管理までの全ライフサイクルにわたり、そこで必要とされる各種作業を支援する考え方、製品を体系化したものである。情報システム開発の計画技法、開発方法論、支援ツール群、およびこれを支えるリポジトリで構成しており(9ページの図参照)、これらが一体となって情報システムの開発を支援する。

(2) マルチターゲット開発環境の実現

多様な情報システム開発のニーズに対応するため、ワークステーション業務の開発からホストコンピュータ中心の大規模システムの開発まで、幅広いアプリケーションの開発が同一の開発環境上で行える。

(3) ソフトウェア一貫生産システム

ソフトウェア開発の上流工程から単体テストまでをワークステーション上で実行できるようにした。また、上流から下流までのソフトウェア開発に必要な情報資源を共有化・一元管理を可能とするリポジトリサービスにより、ツール間の結合を強化している。

2.2 CAPSDFの構成

CAPSDFは、技法・方法論・支援ツール・リポジトリが一体となって情報システムの開発を支援する。個々の構成要素について以下に述べる。

(1) 情報システム統合計画技法“HIPLAN”

情報システム統合計画技法HIPLAN(Hitachi Integrated Planning Procedure for Information System)は、経営に寄与する情報システムの計画・分析をユーザーの視点に立って具体化する計画技法である。情報システムの戦略の企画“HIPLAN-SV(Strategic System Vision)”, 情報システム全体計画の立案“HIPLAN-MP(Master Plan)”, 個別システム開発計画の策定“HIPLAN-AP(Action Program)”, 個別システムの外部仕様の決定“HIPLAN-AD(Application Design)”という4ステージで構成しており、企業全体から部門まで、さまざまな情報化のニーズに対応する。

(2) 情報システム開発方法論“HIPACE”

情報システム開発方法論HIPACE(Hitachi Phased Approach for High Productive Computer System's Engineering)は、HIPLANによって計画・分析された情報システムを実現するための開発方法論である。ユーザーニーズの把握から情報システムの設計・開発・テスト・運用・管理まで、ライフサイクルでの一連の作業をフェーズごとにシステム開発標準手順“SPDS”(Standard Procedure to Develop System)として標準化している。

また、開発手順の各作業で使用される技法の主なものとしてデータ中心アプローチ、オンラインプロトタイピング技法などがある。オンラインプロトタイピング技法とは、ユーザーの要求仕様を早期確定するとともに、オンラインプログラムの構造を画面制御とメッセージ処理に分離するよう標準化し、生産性と品質の向上を図る技法である。データ中心アプローチについては3章で詳述

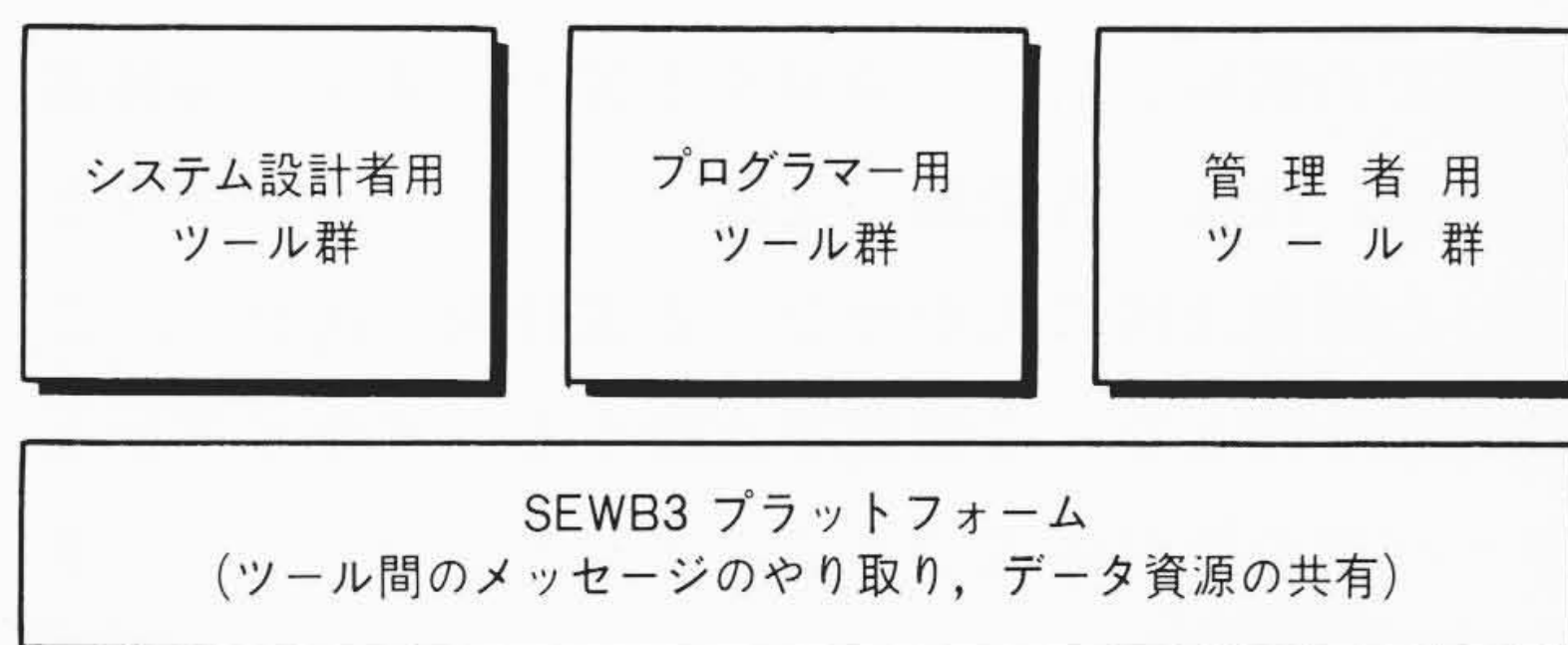


図1 SEWB3ソフトウェアの構成 SEWB3は、システム設計者用ツール群、プログラマー用ツール群、管理者用ツール群およびこれらを搭載するプラットフォームで構成する。

する。

(3) 支援ツール群

CAPSDFでの支援ツール群は、上述のシステム開発計画技法、開発方法論を利用したシステム開発を支援するツール群である。主なツールについて以下に述べる。なお、SEWB3(Software Engineering Workbench 3)、EAGLE/4GL(Effective Approach to Achieving High Level Software Productivity/4th Generation Language)の詳細については、この特集号の別論文「ソフトウェア開発支援ツール“SEWB3、EAGLE/4GL”の機能と特長」を参照されたい。

(a) ワークステーション型統合開発支援環境“SEWB3”

SEWB3は、ワークステーション上の開発支援環境で、システム設計者用ツール群、プログラマー用ツール群、管理者用ツール群、およびこれらを搭載したプラットフォームで構成する(図1参照)。

(b) ホスト型開発支援ツール“EAGLE2”

EAGLE2は、ホストコンピュータ上で動作する開発支援ツールである。今までに培われたソフトウェア開発情報資源を有効に活用しながら、信頼性の高いシステムが効率よく開発できる。また、SEWB3と連携したシステムの開発も行える。

(c) 第4世代言語“EAGLE/4GL”

EAGLE/4GLは、頻出する業務処理に限定して処理をパターン化して基本アルゴリズムの検討を不要とすることにより、プログラミング作業の効率を飛躍的に向上させるものである。さらに、バッチ系として、データベースやファイル内容の一覧表、集計表など帳票出力業務のための自動作表機能(図2参照)、対話系として、一品一葉、多品一葉などきめ細かい対話サブパターンを持つ。

(d) リエンジニアリングツール

リエンジニアリングツールの機能は、既存ソフトウェアを分析して上流の仕様情報を抽出するリバースエンジニアリング、既存ソフトウェアを再利用が容易な形式に変換するリストラクチャリングおよびフォワードエンジニアリングに大別される。リバースエンジニアリングツールは、プログラム仕様書、ファイル仕様書、画面・帳票仕様書などSEWB3、EAGLE2の各種仕様書をソースプログラムから逆生成する。リストラクチャリングツールは、データ標準化機能とプログラム標準化機能を備えている。データ標準化では、既存ソフトウェア内のデータ項目を分析し、その結果をSEWB3、EAGLE2のデータ項目辞書、データ仕様書に登録する。プログラムの標準化では、既存ソースプログラムを、標準データを使用した標準ソースプログラムに書き換

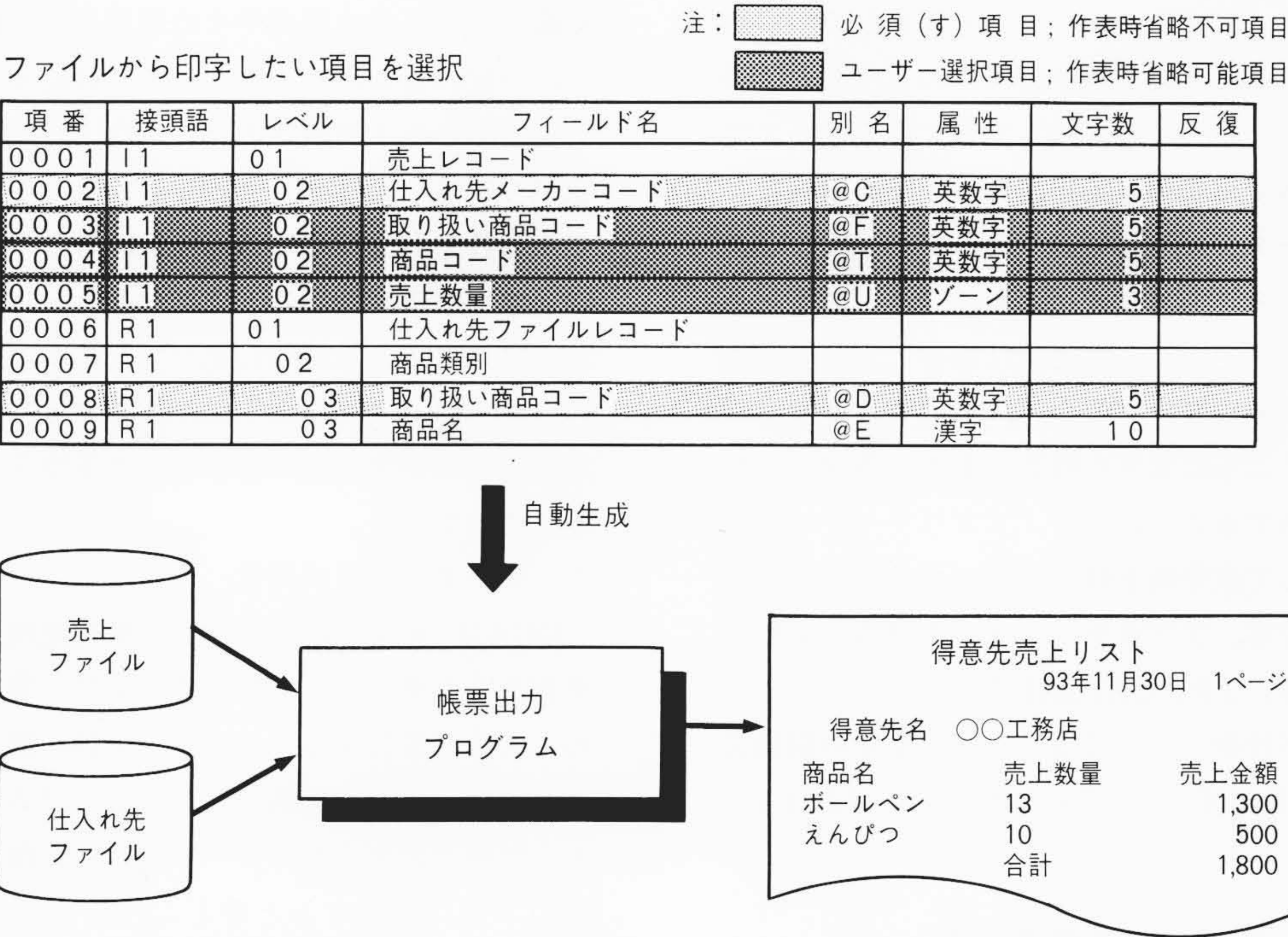


図2 EAGLE/4GL自動作表機能 自動作表機能によってファイルの内容を帳票に出力するアプリケーションを、プログラミング作業なしに開発することができる。

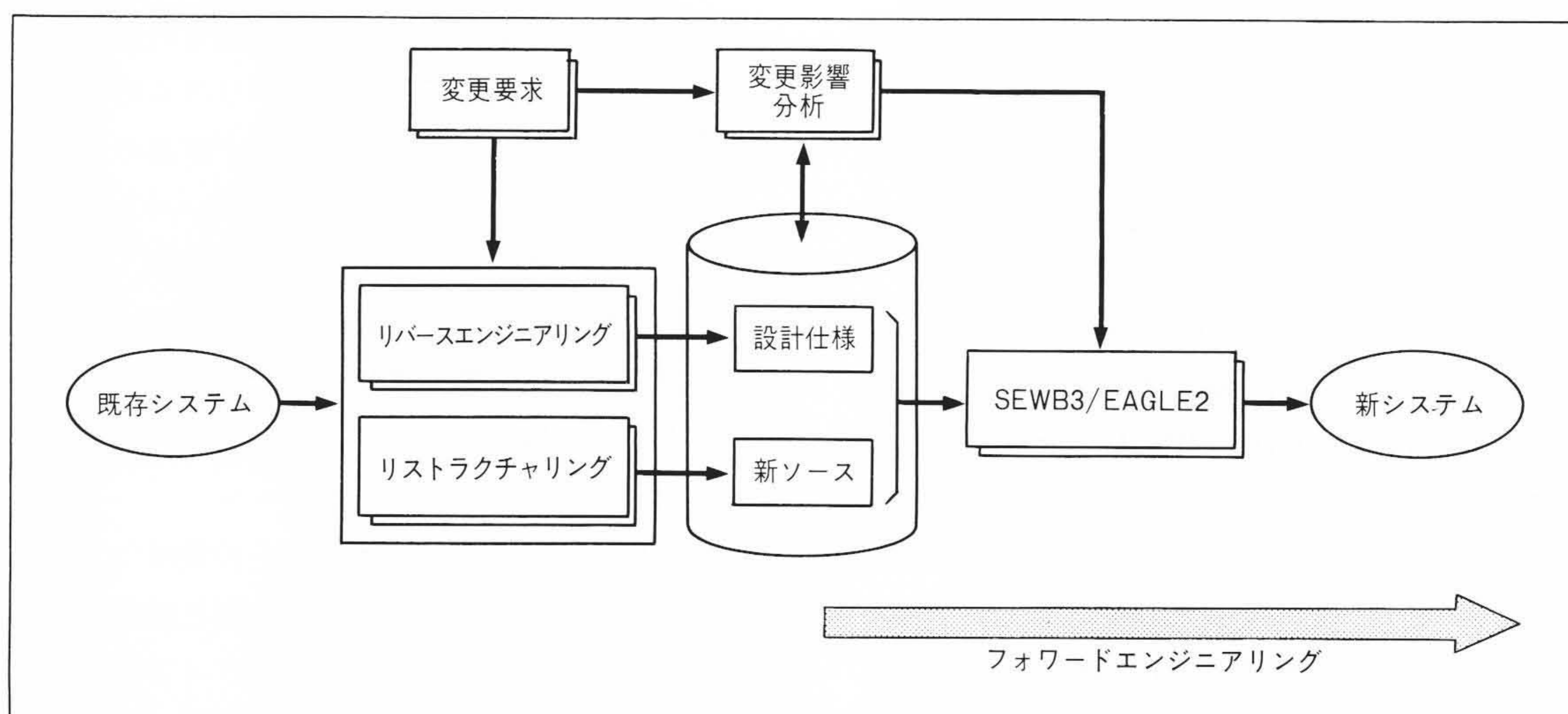


図3 リエンジニアリング リバースエンジニアリングやリストラクチャリングによって抽出された設計情報や、ソースで変更影響分析を行って得られた結果に基づいて、新システムの構築を行う。

える。リエンジニアリングツールの抽出設計情報は、通常のフォワード エンジニアリング ツールSEWB3、EAGLE2などで利用可能である(図3 参照)。

なお、リバースエンジニアリング、リストラクチャリングについてはこの特集号の別論文「ソフトウェア資産を有効活用するリエンジニアリング支援システム」を参照されたい。

(4) リポジトリ(情報資源倉庫)

リポジトリとは、アプリケーションを開発していく過程で生産・使用される上流から下流までのすべての情報資源を、共有化・一元管理を可能とするデータベースである^{3),4)}。情報資源としては、業務用語やデータ項目などの辞書情報、データフロー図やシステムフロー図などの仕様書情報、ソースコードなどがある。これらの情報を単独に管理するだけでなく、例えば仕様書どうしの関連や仕様書とデータ項目間の関連など、情報間の関連情報も管理しているところに大きな特徴がある。各ツールがリポジトリを共有することにより、アプリケーション開発の全工程を通して成果物を統合管理できるので、開発情報の再利用ができ、アプリケーション開発の生産性・信頼性が向上する。なお、CAPSDFでのリポジトリの具体的な説明、利用技術については、この特集号の別論文「ソフトウェア開発支援ツール“SEWB3, EAGLE/4GL”の機能と特長」を参照されたい。

3 データ中心アプローチの考え方

この章では、CAPSDFの主な特徴の一つであるデータ

中心アプローチの考え方について述べる。

DOA(Data Oriented Approach: データ中心アプローチ)とは、「システム開発での最も重要な資源はデータである。」という考えに基づいてデータ分析と標準化を行い、システム設計以降の開発作業に利用する方法のことである⁵⁾。このアプローチの特徴について以下に述べる。

(1) 企業・組織全体が対象

DOAは、企業や組織全体を検討の対象とする⁶⁾。企業や組織でのシステム開発やその運用での問題を解決しながら、一定の目標概念へ導くことを目的としている。すなわち、単なる1回限りのソフトウェア開発だけでなく、企業や組織で繰り返し実施されるソフトウェア開発全体の生産性を向上させることを目的としているのである。ユーザー要求や個別のトランザクションに基づく従来の処理プロセスの設計法に比べて、標準データの安定性、ユーザーのデータ有効活用、データとプログラムの関連単純化、標準データに基づくソフトウェア部品化などの点で優れている。

(2) データの共有資源化

DOAは、企業や組織全体での情報処理の中核にはデータがあると考え、その整合性および一貫性を確保するために共有資源化する。データを中核に置いて考える理由としては、企業や組織が抱えるシステム開発や運用に関する問題点の多くが、データや情報に関する資源管理体制の不備に起因すると考えられる点、企業間の競争激化・顧客ニーズの多様化などに伴い企業での「情報生産」の概念が台頭した点、企業が使用するデータが一般には

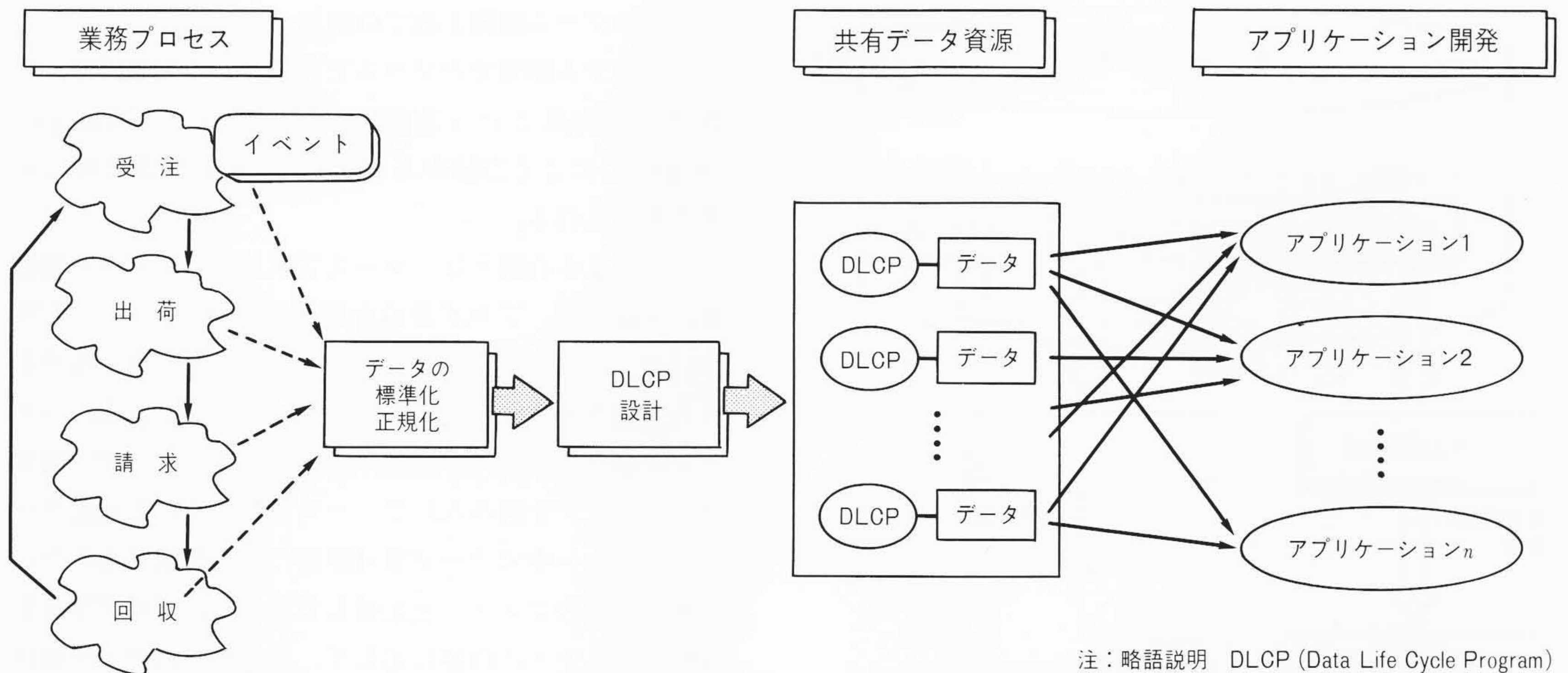


図4 データ中心アプローチとDLCP データ中心アプローチでは、実世界の実体に対する有意なイベントに対応する外部的な操作によってデータをカプセル化する。

それほど変化せず安定しており、これに基づいて構築されるアプリケーション群も安定することが期待される点などがあげられる。

(3) データのカプセル化

DOAでは、企業・組織の保有するデータの共有化を図る際に、あらかじめ正規化・標準化したデータをカプセル化して保持する。データのカプセル化とは、そのデータの実現方法などの内部情報を外から隠して、外部的なインタフェースによるだけでデータを操作させるようにする手法であり、データ抽象とも呼ばれる。データ処理に関するプログラムを簡潔にできると同時に、データの実現にかかわる変更波及がそのカプセルの外に及ばないので、保守性の高いシステム構築が可能となる。

カプセル化したデータへのアクセス手段である外部的な操作は、実世界の実体に対する有意なイベント発生と対応しており、一つの実体の発生から消滅に至る一連のイベントを、データのライフサイクルとして定義する。このようなイベントに対応した処理プロセスを集約・統合したものを、DLCP(Data Life Cycle Program)と呼ぶ(図4参照)。

4 データ中心アプローチに基づく“CAPSDF”によるアプリケーション開発支援

CAPSDFでのデータ中心アプローチの中核は、リポジトリ内で管理されるデータ項目辞書である。データ項目辞書を中心としてその周辺のツール群とのかかわりにつ

いて以下に述べる(図5参照)。

4.1 データ項目辞書

データ項目辞書は、情報システム開発で使用するデータを保管、管理する辞書であり、前章で述べたデータ中心アプローチに基づいて抽出されたデータ資産の格納場所とすることができる。リポジトリ内で他の仕様書類と同じように管理され、データの一貫性の保持、保守性の向上、再利用の促進、システム開発の効率向上などを目的としている。この辞書の内容はデータ項目と業務用語に大別される。データ項目はプログラムソースコード上で使用されるものであり、項目名や属性(型、けた数など)を持つ。業務用語はSDF(Structured Data Flow Diagram)など上流の仕様書で使用されるものであり、処理サイクル、実行部署などの情報を持つ。データ項目、業務用語ともにデータ項目設計支援ツールによって辞書に登録できる。

データ項目辞書にはデータ項目に対する処理をいっしょに登録することができる。登録できる処理には、入力項目チェック処理、入力項目編集処理、出力項目編集処理などがある。これらの処理は、前章でデータ中心アプローチの説明で触れたDLCP、すなわちデータのカプセル化を実現するものということができる。

4.2 データ項目辞書の利用

(1) システム設計工程での利用

システム設計支援ツールを使用して、画面・帳票、レコード仕様、リレーショナルデータベースの表仕様など

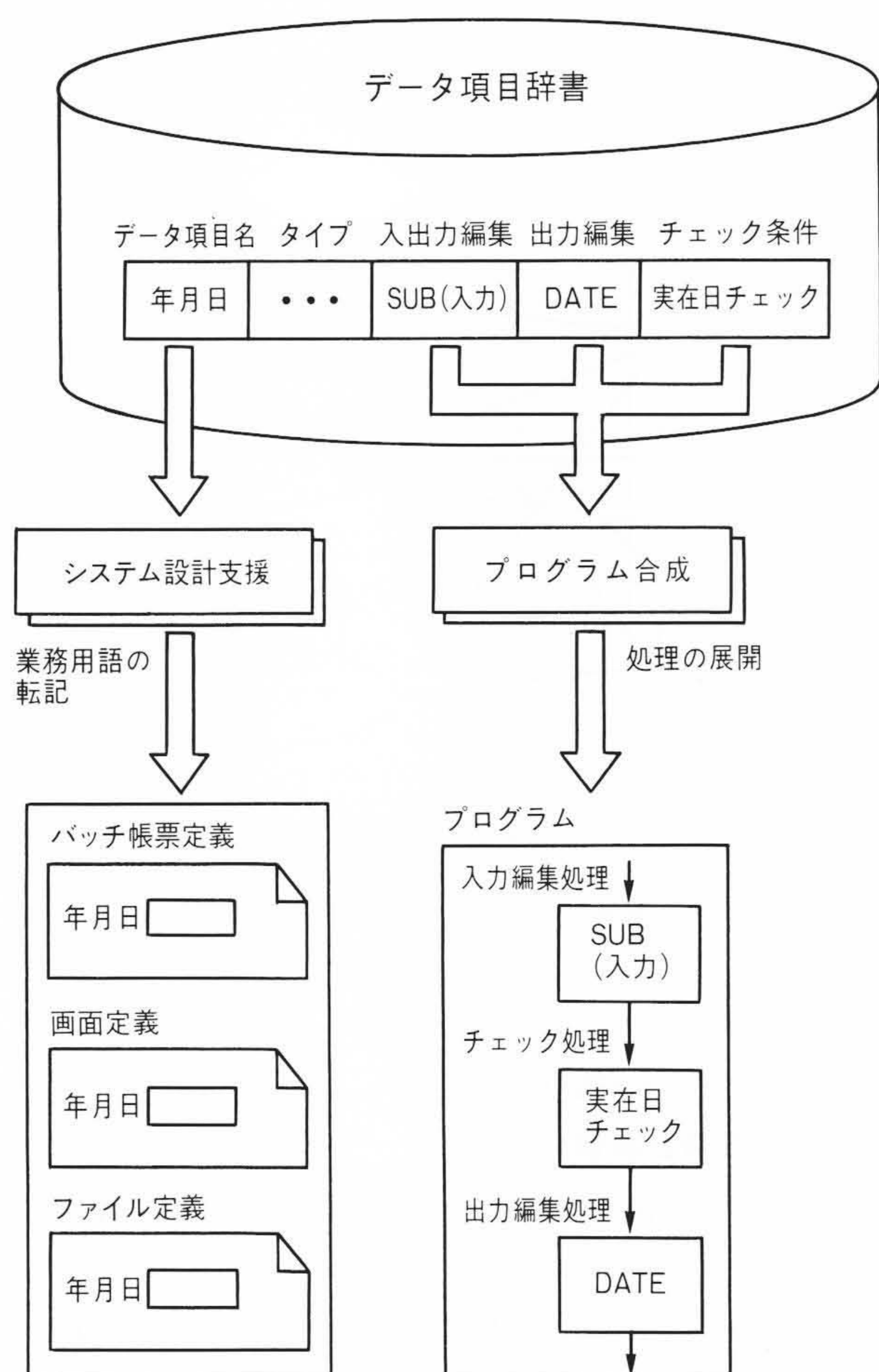


図5 データ項目辞書とツール群とのかかわり合い データ項目辞書は、システム設計では業務名の転記に、プログラム合成ではデータ項目ごとに割り付けられた処理のプログラム中への展開に利用される。

を定義する際に、この辞書からデータ項目や属性などを転記して使用することができる。データの再入力の手間を省くことができると同時に、システムで使用するデータの名称や属性を統一的に扱うことによって一貫性を持ったデータの体系にでき、標準化された内容の仕様書を作成できる。

(2) プログラム開発工程での利用

プログラム開発でのソースプログラムの生成には、プログラム合成という方法とPAD(Problem Analysis Diagram)による方法がある。データ項目辞書は特に前者で利用される。

プログラム合成とは、ソースプログラムをデータ定義部、手続き部、プログラムを形作る骨格(パターンと呼ぶ。)の三つの部分に分けて別々に用意しておき、後でそれらを組み合わせて完成させるという方法である。パターンが形作る骨格に、部品や定義仕様書、ユーザー追加コーディングを組み入れてソースプログラムを完成させる。パターン中のデータ項目辞書の内容を取り込みたい場所に独自のコメントを記述しておく、プログラム生成時にコメントの内容に応じて、あらかじめデータ項目辞書に登録した入力項目チェック処理、入力・出力編集処理をパターン上に組み込む。

なお、データ項目辞書の構成方法や利用方法については、この特集号の別論文「データ中心アプローチを応用した標準データ項目辞書の開発とその活用方法」を参照されたい。

5 おわりに

ここでは、アプリケーション開発支援体系CAPSDFの全容について、技法・方法論・ツールというその構成要素の特徴を個別に論じただけでなく、データ中心アプローチという考え方を中心としたこれらの協調関係について述べた。

この体系ではこのほか、オンラインプロトタイピング技法、構造化技法、部品再利用技法、プロジェクト管理技法などを技法方法論としてサポートしている。データ中心アプローチと同様、これらの技法もツールと緊密につながりがあるということをひと言加えて、この論文の結びとしたい。

参考文献

- 1) 佐藤：CASEツール，(株)ソフト・リサーチ・センター(1990)
- 2) Martin, J.: Information Engineering Book I, Prentice-Hall, Inc. (1989) 竹村訳：インフォメーションエンジニアリングI，トッパン(1991)
- 3) 山田：情報資源管理概論，オーム社(1987)
- 4) 酒井：情報資源管理の技法，オーム社(1989)
- 5) 堀内：データ中心システム設計，オーム社(1988)
- 6) 堀内：企業情報システムにおけるデータ中心手法導入の要件，情報処理学会論文誌，Vol.33, No.4, 521～531(1992)