

SEWBを用いたソフトウェア開発環境の整備

—日立製作所の計画—

Improvement of Software Development Environment using “SEWB”
—Hitachi’s Plan—

企業のコンピュータシステムは、情報伝達のスピードアップと経営の軽量化などの経営ニーズに対応してますます高度化・拡大化している。このような環境下において、エンドユーザー部門での機械化・合理化要求の急増に伴う膨大なシステム開発作業量は情報処理機械化部門の開発能力をはるかに上回り、ますますバックログ(開発待ち業務)を増大させる傾向にある。

このような背景を踏まえて、日立製作所情報管理部門では業務の標準化と一体となった「システムとソフトウェアの標準化」を全社を挙げて推進中である。また、開発過程での「ソフトウェアの生産性向上」を図るためHIPACEの標準手順・技法を一体化した開発支援ツールEAGLE2/SEWBを活用中であり、開発工数の大幅削減を目指している。

大貫政男* Masao Oonuki
坪井星吉* Seikichi Tsuboi
今川弘道* Hiromichi Imagawa
柴崎浩美* Hiromi Shibazaki

1 緒言

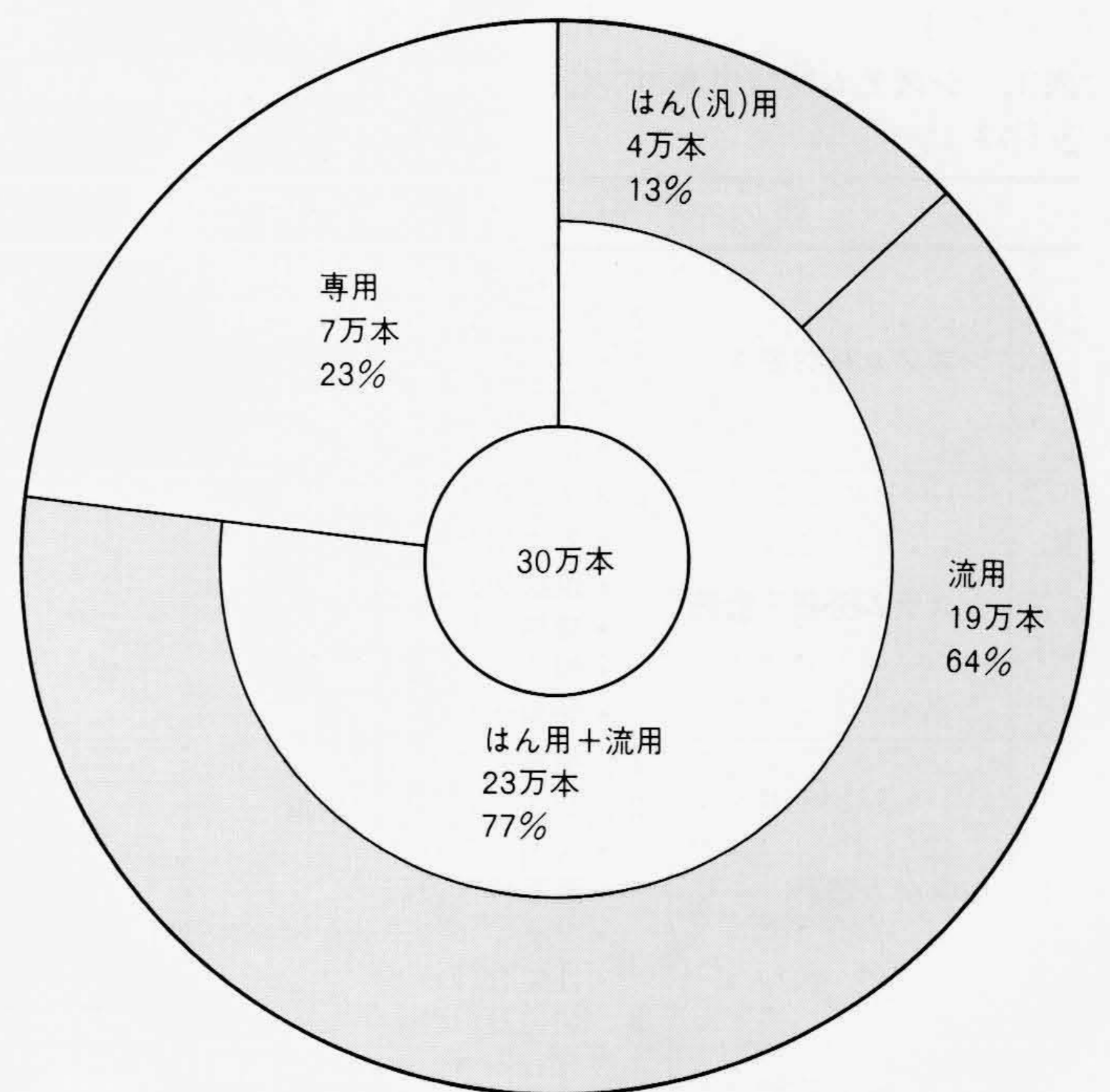
間接業務合理化のシステム開発を担当している日立製作所情報管理部門は、ここ10年来人員が横ばいであるが、ホストコンピュータ設置台数の倍増や保有プログラム本数の4倍化による作業増によって、ソフトウェアの生産性向上が急務となっている。加えて、システムの大規模化と複雑化により、その信頼性も重要となっている。

本稿では、ソフトウェアの標準化(以下、ソフト標準化と言う。)の推進の観点から、EAGLE2¹⁾/SEWB^{2),3),4)}(Effective Approach to Achieving High Level Software Productivity 2/Software Engineering Workbench)の適用計画によるソフトウェア生産性向上について概説する。

2 ソフト標準化の推進

2.1 計画の概要

本計画は業務の標準化を行い「最小の費用で最大の効果を挙げられる経営体質」の早期実現をねらいとしているが、情報管理部門が抱えているシステム開発のバックログ解消の切り札ともなる対策である。情報管理部門で稼動している事務系プログラム本数は昭和62年現在約13万本であるが、ユーザーニーズに対応して今までどおりソフトウェア開発を進めると30万本にもなると予想される。ソフト標準化計画は、**図1**に示すとおり30万本をニーズに対応し各部署が個別開発・専用利用するのではなく、できる限り全社共通システム化をねらいとして共同開発・共同利用を図りはん(汎)用プログラム化を促進しようとするものである。はん用プログラム(4万本)



昭和66年3月計画

注：はん用(複数事業所・事業グループ・全社共通のプログラム)
流用(事業所間で共同利用したプログラム)
専用(事業所専用のプログラム)

図1 ソフト標準化計画 はん(汎)用プログラムの共同開発と共同利用(流用)により、専用プログラムの開発を大幅に削減する。

* 日立製作所情報管理部

を全社的に共同利用することによって、専用プログラムを7万本以下に削減する計画である。この結果、ソフト標準化率は「{(はん用プログラム+共同利用プログラム)/全稼動プログラム} × 100%」の定義で求めると、現在37%から昭和66年には77%を目指している。

2.2 ソフト標準化計画の基本的な考え方

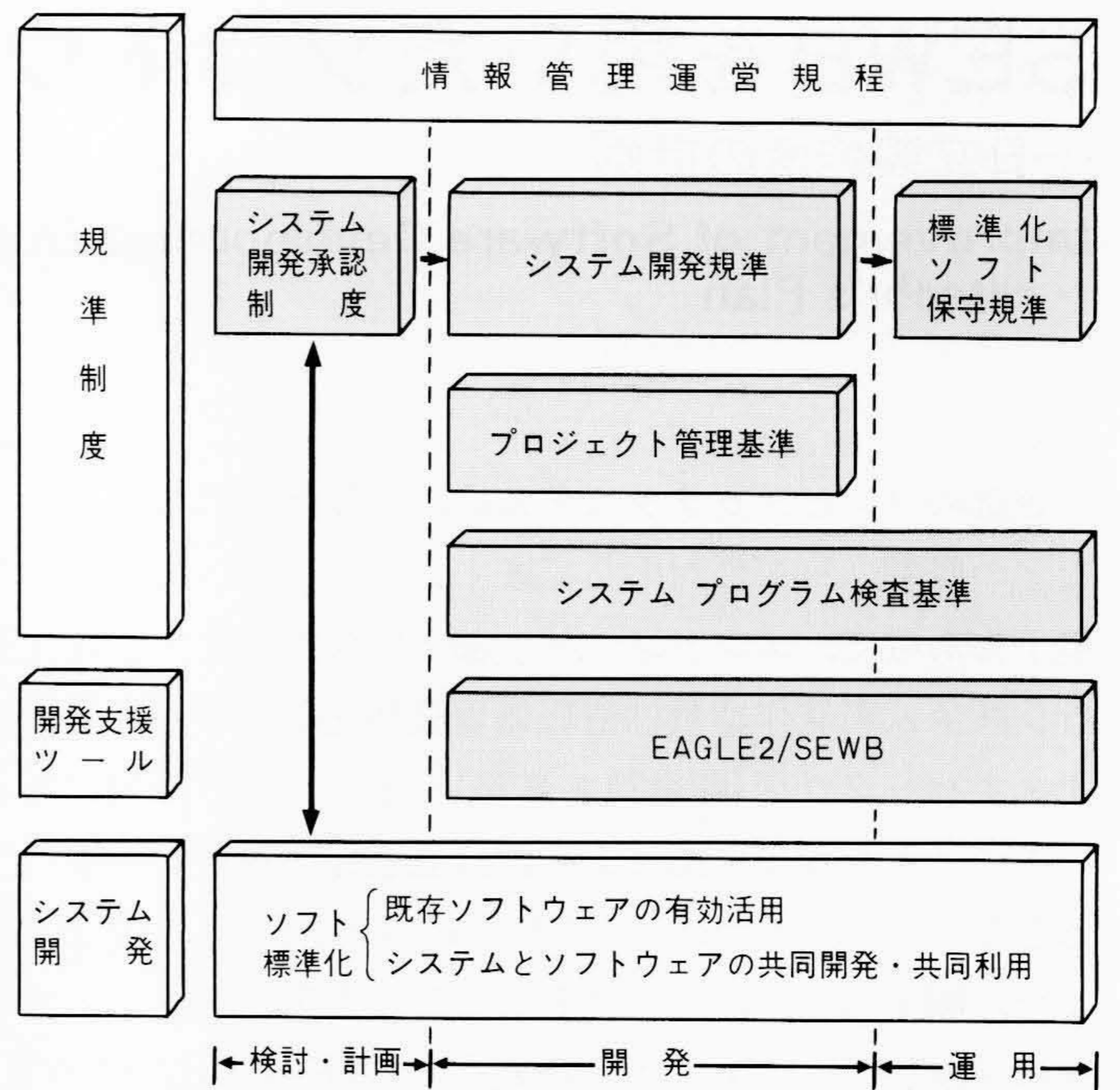
ソフト標準化を進める上での基本的な考え方を以下に示す。

- (1) 既存ソフトウェアの有効活用と共通システム・共同開発を推進する。
- (2) 新規システムは、全社共通又は事業所共通システムを前提として開発する。
- (3) ソフト標準化の前提条件として業務の標準化をエンドユーザー部門のリーダーシップにより強力に推進する。

2.3 ソフト標準化の基盤整備

ソフト標準化の推進に当たり、システム開発の基盤となる規準と制度を整備中である。主な規準を図2に示す。その考え方は以下のとおりである。

- (1) システムの重複開発を防止する「システム開発承認制度」
- (2) システムの開発手順及びドキュメントの作成方法を標準化する「システム開発規準」(表1)
- (3) 共同開発した標準化ソフトの受渡し方法や保守責任元をルール化する「標準化ソフト保守規準」
- (4) システム開発時のプロジェクトマネジメント手法を標準



注：略語説明

EAGLE2/SEWB (Effective Approach to Achieving High Level Software Productivity 2/Software Engineering Workbench)

図2 システム開発に関する制度・規準などの整備 システム開発承認制度による重複開発の防止とソフトウェアの生産性・保守性・信頼性向上が基本となる。

表1 システム開発規準の整備 システムの共同開発・共同利用に伴い、生産性・保守性・信頼性の観点から工程別の作業内容と手順を標準化した。

工 程	作 業 内 容	特 に 考 慮 し た 点
1 システム検討要求	1. システム検討依頼書作成 2. システム要求仕様書作成 • 現状業務 • 改善後業務 • 推進体制, 日程, 効果	• 「システム開発承認制度」による (1) 既存ソフトウェアの有効活用の促進(重複開発の防止) (2) 共同開発・共同利用の促進 • 共通システムの開発は開発体制, 規模, 工数, 日程を明確化しオーソライズする。
2 システム分析・計画	システム基本計画書作成 • 基本構想 • 現状業務調査分析 • 業務仕様の設定 • 類似システム調査 • 開発体制, 日程, 効果	• 2~6の工程については大・中・小日程を作成し, 定期的にフォローを行う。 • 各工程の終了時点でデザインレビューを行い, 次工程へ進む。
3 システム設計	1. 業務処理内容の決定 2. システム仕様書作成 • 入出力設計 • 画面帳票設計 • ファイル, データベース設計 • 性能設計など 3. 開発日程表	
4 プログラム設計	1. プログラムモジュール設計 2. プログラム仕様書作成	• モジュール分割による信頼性向上 • プログラムチェックリストの作成
5 プログラム作成	1. コーディング 2. 単体テスト	• コーディングルールの設定 • 構造化プログラミング
6 テ ス ト	1. 機能の検証 2. 性能の検証 3. 運用, 操作の確認	• 総合テストによる機能・性能の検証 • 品質の定量的把握 • テスト結果のレビュー
7 運用・保守	1. 使用環境の変化に応じた修正・改良 2. 完成システムの使用と誤りの補修 3. 運用マニュアルの作成	• 「標準化ソフト保守規準」による共同利用するシステムの統一性維持 • エンドユーザーへの教育徹底
8 その他全般として	1. データ名称, プログラム名称の付け方 2. ドキュメントの記述内容 3. プロジェクト開発の標準手順書	• 全社統一コード体系の設定 • 標準ドキュメント様式の設定 • プロジェクトマネジメントに必要な様式・ドキュメントの設定

的にする「プロジェクト管理基準」

(5) システムやプログラムの品質・信頼性を確保するための検査基準

3 EAGLE2/SEWB適用計画

3.1 開発支援ツールの整備

システムの共同開発には、共通の開発環境の整備が必要である。そこでプログラムの標準化、パターン化、部品化を図る武器として、開発支援ツールEAGLE2/SEWBを採用し全社標準ツールとして全事業所に展開する計画である。適用計画で、EAGLE2について昭和61年度に社内モデル5事業所によって試行と評価を行った(表2)。評価の結果、同表に示すとおりプログラム作成で2倍、画面定義では5倍、テストでは1.3倍の生産性向上を確認し、昭和62年度から全社展開中である。また、SEWBについては、昭和61年度に製品化に先立ってプロトタイプを試行評価し、使い勝手などの改善要望事項を製品開発へ反映した。昭和62年度に社内のモデル8事業所が製品版SEWBを使用して、システム設計からテストまでの一貫したシステム開発手順(図3)について試行し、運用方法を定めた上で昭和63年度から全事業所への展開を図る予定である。特に、システム分析・計画、システム設計などシステム開発の上流工程やデバッグ テストなどの検査工程にSEWBを積極的に導入し、情報管理部門のSE(System Engineer)、プログラマのトータルの生産性を2倍以上にする計画である。SEWBの設置基準はSE3人/台、SE兼プログラマ2人/台、プログラマ1.5人/台を考えている。

また、事務系ソフトウェアの開発での作業内容とEAGLE2/SEWBの関係を表3に示す。システム分析・計画工程では調査分析・思考が主体であるためAI(Artificial Intelligence)技術を応用した支援ツールの開発が望まれる。EAGLE2/SEWBは表3に示すとおりシステム設計以降の工程をほとんどサポートしている。システム分析・計画からテストまでの各工程で作成するドキュメントの60~70%が機械化される。

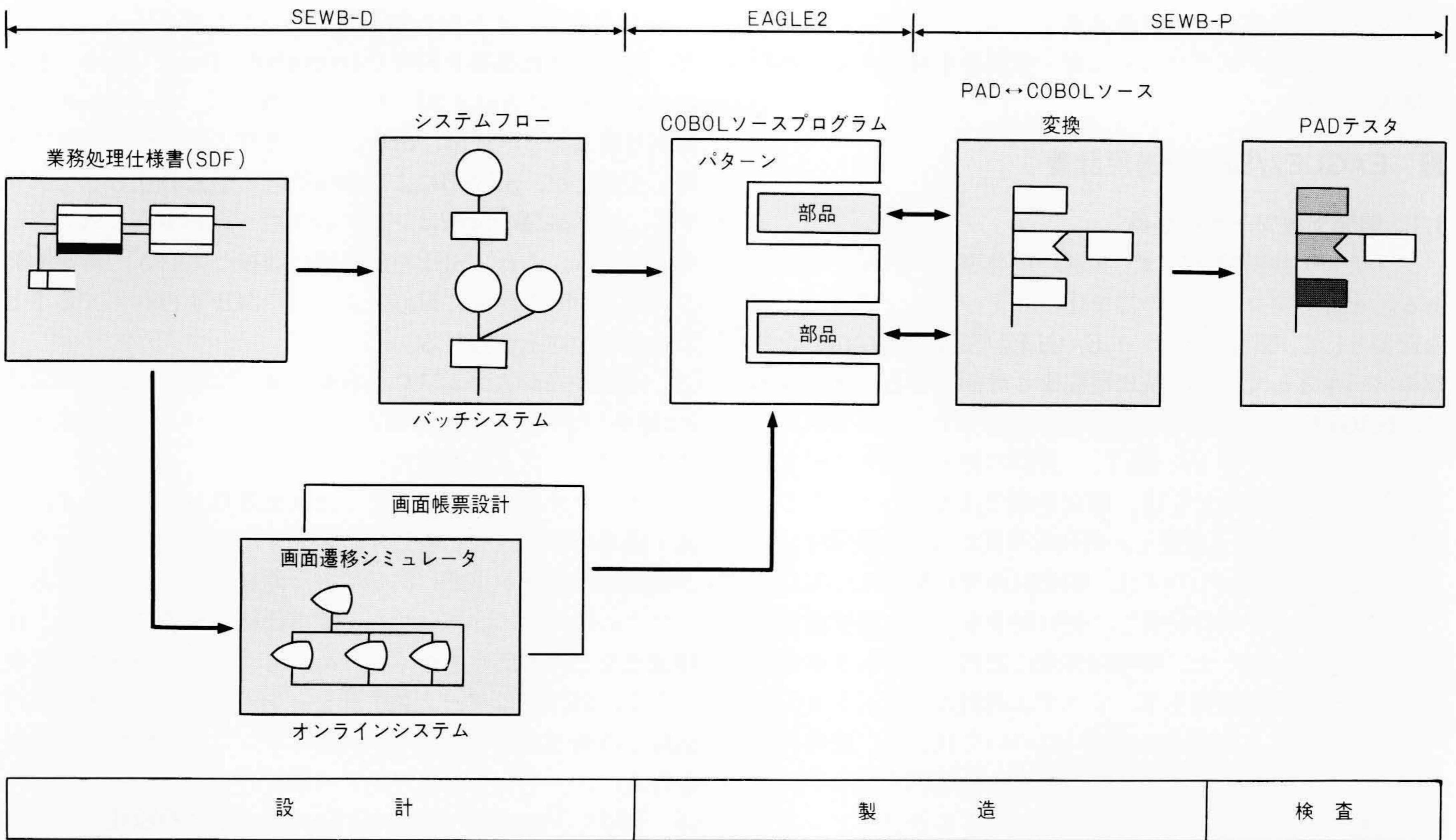
SEWB適用による主な効果は次のとおりである。

- (1) 業務処理仕様書をSDF(Structured Data Flow Diagram)で書くことにより^{2),3)}仕様書の再利用、後工程へのつながりが可能となる。従来、SDFは業務処理の整理技法として利用していたが、SEWBにより機械処理の設計技法として利用する。機械処理のSDFはシステムフロー自動生成の入力として活用する。また、SDFを計算機に蓄積しておき、新規開発システムのSDF作成に類似システムのSDFを再利用することで生産性の向上が図れる。
- (2) SDFからシステムフロー自動生成^{2),3)}の過程で、EAGLE2は標準パターンで構成されるシステムフローを自動生成するので、プログラムの標準化が図れる。
- (3) オンラインシステム設計では画面遷移シミュレータ、画面・帳票設計支援などのプロトタイピング^{2),3)}によりオンライン機械処理仕様を早期に決定でき、開発の効率が向上する。システム開発時にはユーザーの要求仕様を早期に決定し、仕様変更などの手戻りをなくすためにもユーザーの参画が必要である。SEWBはワークステーション上で画面・端末帳票の設計と画面遷移シミュレータによるオンライン機械処理設計が行え、ユーザー自身がシステム設計に参画できる。
- (4) PAD⁵⁾(Problem Analysis Diagram)→COBOLソース変換^{2),3)}により、図形によるプログラム作成と構造化された部品の開発ができ、ソフトウェアの品質・信頼性が向上する。また、COBOLソース→PAD変換により仕様書とソースプログラムが一致し生産性・保守性が向上する。
- (5) PADテスト^{2),3)}によりテストの十分性を視覚で確認できる。プログラムやモジュールの品質が大幅に向上し、システム全体の信頼性向上が図れる。
- (6) SEWBによりホストコンピュータに拘束されることなく、ソフトウェアの開発ができる。また、分散処理による応答性の大幅な改善によりシステム全体の生産性が向上する。
- (7) EAGLE2/SEWBの適用により表3に示すような設計仕様書の機械化によるデータベース化が実現し、仕様書の再利

表2 EAGLE2評価結果 EAGLE2の適用によって、プログラム作成の生産性が2倍に向上した。SEWBを適用すれば更に上流工程の生産性向上とシステムの信頼性向上が期待できる。

工 程	量的効果 (従来工数を1として)	質 的 効 果	備 考
1 システム設計	画面・帳票定義 0.2 書式定義 0.2	—	SEWB-Dに期待
2 プログラム設計	パターン・部品による プログラム設計 0.5	仕様書の標準化 主管元での設計	SEWB-Dに期待
3 プログラム作成	0.5	パターンによるプログラムの標準化 信頼性向上	標準パターン・部品の適用。更にSEWB-Pによる生産性向上期待
4 テ ス ト	0.7	信頼性向上	更に、SEWB-Pによる信頼性向上期待
5 保 守	調査工数 0.75 (関連プログラム) (ファイルなど)	ドキュメントの整備 ADCAS: 8種類 EAGLE2: 4種類	標準パターン・部品の適用と機械出力ドキュメントの利用による保守性向上

EAGLE2
対象範囲



注：略語説明 SDF(Structured Data Flow Diagram), PAD(Problem Analysis Diagram)

図3 EAGLE 2/SEWBによる事務系ソフトウェア一貫生産システム システム設計からテストまでを一貫して開発する。SEWBの適用による設計工程の生産性向上と検査の信頼性向上が図れる。

用によるソフトウェアライフサイクルでのソフト生産性が向上する。また、新規システムの開発で、設計仕様書データベースから類似システムの設計仕様書を検索して再利用するなど、ソフトウェア開発の大幅な生産性向上が可能となる⁹⁾。

3.2 既存プログラムの保守、再利用システムの開発

新規開発では、EAGLE2/SEWBの適用によって生産性・保守性・信頼性が大幅に向上する。情報管理部門は前述のように常に大量の既存プログラムを保有しており、その保守の効率化が重要となっている。そこで既存プログラムを有効活用

表3 開発作業の内容と支援ツール 事務系ソフトウェア開発でのSEWBとEAGLE 2の機能分担を示す。

開発工程	作業内容	支援ツール	
		SEWB	EAGLE 2
1	システム分析・計画	1. ニーズ分析	-
		2. データ分析	-
		3. 業務分析(SDF)	○
2	システム設計	4. 業務処理設計	○
		5. システムフロー設計	○
		6. 入出力設計	-
		7. 画面・帳票設計	○
		8. ファイル設計	-
3	プログラム設計	9. データベース設計	○
		10. プログラム構造設計	-
		11. モジュール設計	○
4	プログラム作成	12. コーディング	○
5	テスト	13. 単体テスト	○
		14. 連動テスト	-

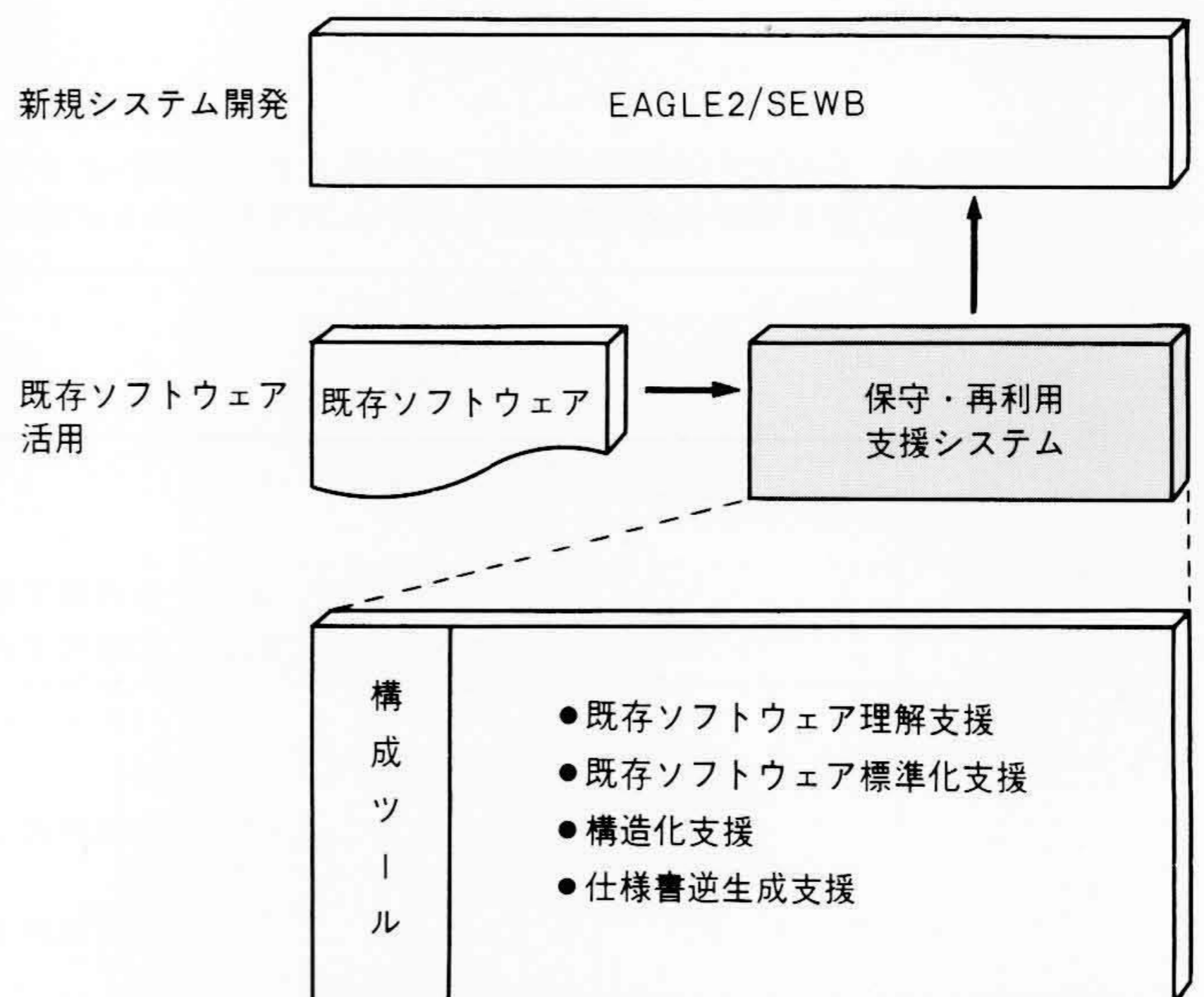


図4 既存ソフトウェアの保守・再利用支援システムの概要 大量の既存ソフトウェアの保守作業を効率化するには、EAGLE 2システムの標準化形式に合わせて標準化することが有用である。

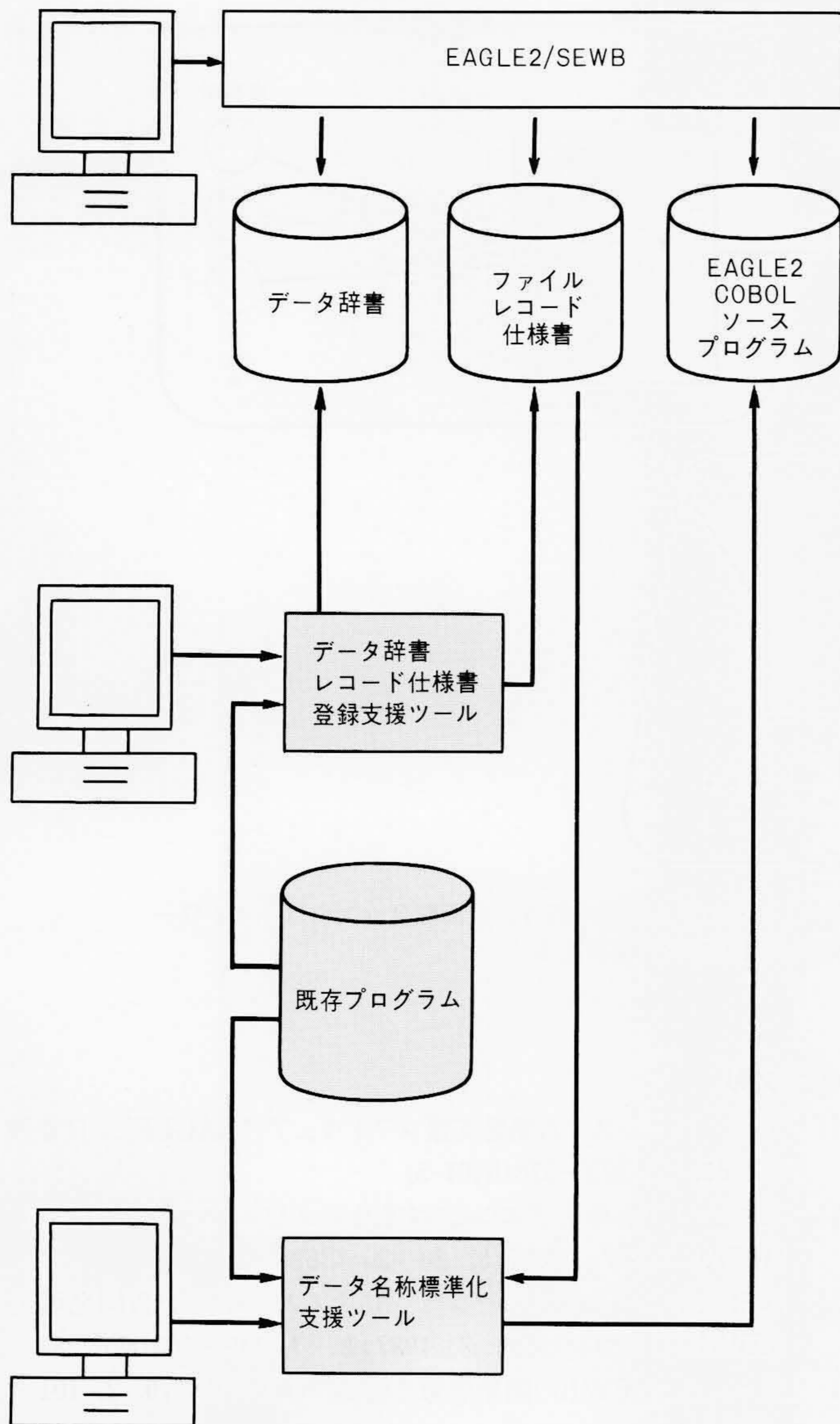
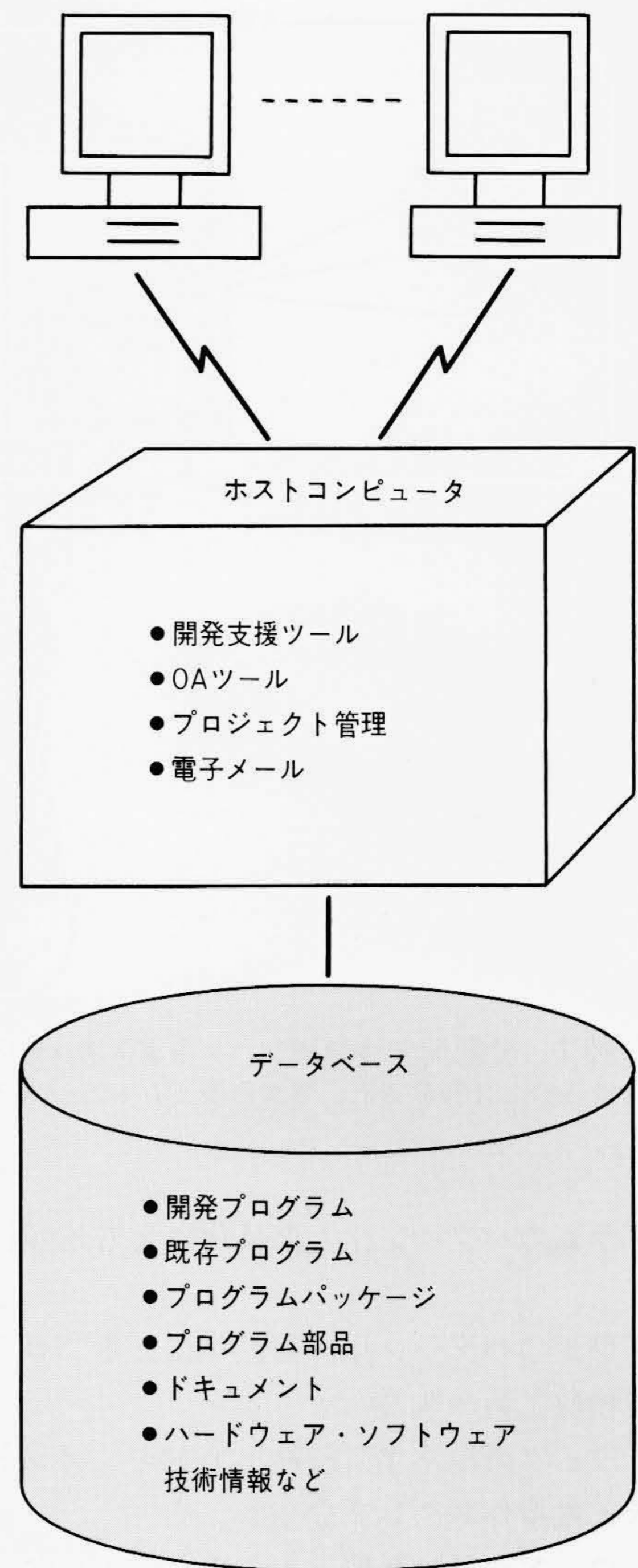


図5 既存プログラム標準化支援ツール 既存プログラムをデータ中心に標準化することによって、保守の効率化を図ることができる。

するには、EAGLE2への移行が必要であり、そのためのツールをシステム開発研究所と共同で開発推進中である(図4)。現在、既存のプログラムからEAGLE2のデータ辞書、レコード仕様書への登録ツールと既存プログラムのCOBOLソースを、標準データ名称のCOBOLソースに変換するツールを開発しており、EAGLE2への移行を容易にしている(図5)。

3.3 ソフトウェア情報のデータベース化計画

現在、社内全事業所の保有プログラム情報、プログラムパッケージ、部品の情報、開発支援ツールの情報などのデータベース化を実施している(図6)。特に各事業所で開発されたはん用性の高いプログラムをデータベースに登録し、事業所相互に利用できるようにしている。今後はシステム開発での分析・計画からテスト・運用に至るまでの各種情報やドキュメントをデータベースに蓄積して、総合的な生産性向上を図る計画である。また、多数の開発要員を1箇所に集めて共同



注：略語説明 OA(Office Automation)

図6 ソフトウェア情報のデータベース化構想 システム開発にかかわる各種情報をデータベース化して、システム分析・計画から運用・保守までのトータルの生産性・保守性・信頼性を向上させる。

開発するには、コスト的にも資源的にも限界があり人事管理上の問題も発生してくる。そこで、共同開発・共同利用を更に促進するために、共同開発センタの設置とネットワークを利用した分散開発体制(図7)を検討中である。

4 適用効果

昭和61年からEAGLE2/SEWBの使用を開始し、現在全社への適用を推進中であるが次の量的効果を確認できた。

- (1) プログラム作成では、従来に比べ約2倍の生産性向上が実現した。
- (2) 設計者用SEWBでは、業務仕様やシステムフローなどの設計仕様の作成工数が、従来に比べ30~50%効率向上した。
- (3) 設計・製造・検査の各工程で作成する60~70%のドキュメントが機械化できる。

一方、定性的効果として次の結果が得られた。

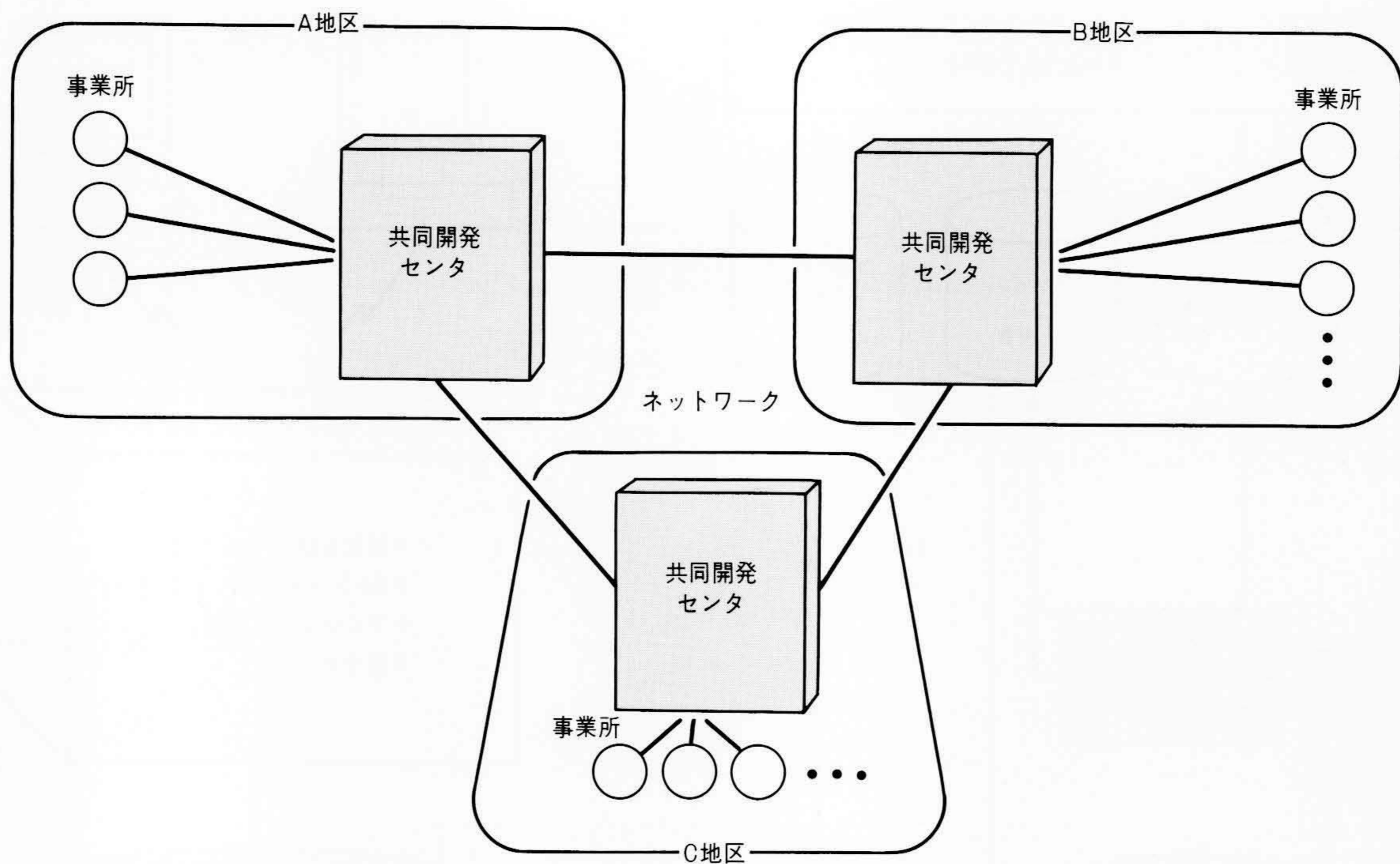


図7 分散開発体制構想 各事業所は地区の開発センターを利用してシステム開発を行う。共同開発センターはネットワークを介して接続され、事業所をまたがるシステムの共同開発・共同利用が効率的に行える。

- (1) プログラムのパターン化と部品化による信頼性・保守性の向上
- (2) プログラムのパターン化、部品化によるプログラム検収の容易化と検収工数の削減
- (3) ソフトウェアの標準化、機械出力ドキュメントの充実によるシステム維持作業の効率向上
- (4) ソフトウェアの事業所間流用の促進と重複開発の防止
- (5) 開発期間の短縮によるシステムの早期立上げ

5 結 言

以上、日立製作所でのソフト標準化の推進と開発支援ツールEAGLE2/SEWBの適用計画について概説した。現在情報管理部門の総力を挙げて推進しているソフト標準化活動は、先行テーマが順次稼動を開始しているところであり、それに伴いアプリケーション ソフトウェアのパッケージ化・部品化と蓄積を行うところである。

また、SEWBのように利用者の身の回りの開発支援ツールについては、開発部署とユーザーとが連携して使い勝手の良いツールにしていくことが重要であり、今後は開発支援ツールを活用していく過程でのノウハウを蓄積して、ソフトウェア生産性の飛躍的向上を図りたい。

参考文献

- 1) 葉木, 外: システム開発支援ソフトウェア“EAGLE”, 日立評論, **68**, 5, 373~378(昭61-5)
- 2) 石本: ソフトウェアエンジニアリングワークベンチSEWB, HITACユーザ, **No.273**, 20~25(昭62-4)
- 3) 津田, 外: ワークステーションによるソフトの分散開発環境 SEWB, 日経コンピュータ, **1987. 8. 17**, 149~160(昭62-8)
- 4) 葉木, 外: SEWBの開発思想と機能, 日立評論, **70**, 2, 101~108(昭63-2)
- 5) 二村: PADの開発, 日立評論, **68**, 5, 351~355(昭61-5)
- 6) 千吉良, 外: システム仕様書再利用によるソフトウェアの開発技法(ICAS-REUSE), 日立評論, **69**, 3, 249~254(昭62-3)