

エキスパートシステム構築標準手順“ESGUIDE”

Methodology for Building Expert Systems

知識工学に不慣れなシステムエンジニアがES (Expert System) を構築するには、ガイドラインとなるべきものが必要である。日立製作所は、今までのES開発経験から得たノウハウをまとめた、ESの開発方法論を提供してきた。開発は4種類の作業フェーズ(構想、計画、開発、運用)で構成され、各フェーズでの作業を誘導する手段としてワークシートを用意している。そして、専門家の知識を知識ベース化するために、専門家モデル、知識モデルと呼ぶ2種類のモデルを導入し、プロトタイピングアプローチを採用することを提案している。さらに、ES単体の開発効果にとらわれることなく企業情報システムの一部として、長期ビジョンにたって設計すべきであることも提案している。

花岡かほる* *Kahoru Hanaoka*
辻 洋** *Hiroshi Tsuji*
丸岡哲也** *Tetsuya Maruoka*

1 緒 言

近年ES (Expert System) 構築ツールの商用化とともに、本格的に実用を目指したESの構築¹⁾が盛んに行われている。一般に、ESの構築には、知識工学に精通したナレッジエンジニアが必要と言われている。しかし、新しい技術である知識工学に精通したエンジニアはあまりいないし、簡単には増えない。したがって、従来コンピュータシステムを開発しているシステムエンジニアが、ESの開発を担当することになるケースがほとんどである。

日立製作所は、知識工学に不慣れなシステムエンジニアでも本格的な実用ESの開発を可能にする開発方法論を、ES構築標準手順ESGUIDE (Expert System Building Guidance for System Designing) としてまとめ、昭和62年から提供してきた²⁾。そして、現在までさらに数多くのESが開発され、その開発ノウハウが蓄積された。このノウハウを従来のESGUIDEに反映させて改良を加え、ESGUIDE第2版を発行した^{3), 4)}。主な改良点は、プロトタイピングアプローチをより具体化し、さらに専門家モデルと知識モデルを作成することにより、知識ベース化がより迅速、かつ正確にできるようにしたことである。

2 ESGUIDEの基本的考え方

ESGUIDE第2版は、次の4項目の考え方に基づいてまとめた。

2.1 システム設計の第一歩は専門家モデルの分析

従来、コンピュータシステムを開発するときにシステム設

計を行っていた。システム設計とは、実世界をモデル化⁵⁾して、計算機で何らかの処理を可能にすることである。「実世界のモデル化」というと非常に難しいことを行うように思われるかもしれないが、実はコンピュータプログラムを開発するとき、いつも行っていることである。ESも、一つのコンピュータシステムであり、コンピュータ化が要求されるものを、コンピュータで実行できる形にすることである。そして、システム設計がやはり必要になってくる。

従来、情報処理システムを設計する場合は、「実世界のモデル化」とは、業務分析であった。そこでは、帳票、台帳、伝票などのデータとそれらの関連を分析して、どの業務を計算機にのせるかを決定するのがモデル化であった。そのための手法として、データフローダイアグラム、ER (Entity-Relationship) モデルなどが利用されてきた。

これに対してESを設計する場合は、「実世界のモデル化」とは、専門家の行う問題解決方法の分析である。すなわち、データ中心の設計から知識中心の設計に変わったと考えてよい。データと知識の区別はあまりはっきりしていない。データも知識の一種ではないかと言われれば、それは否定できない。あまり堅苦しく考えないで、データは「それが持つ値そのもの」に価値があると考え、知識は「推論によって使われて初めて価値が出るもの」と考える。

さらに、従来のソフトウェア開発は、ユーザーの要求(what)を満足するプログラムを実現する(how)ことであると言われていた。ESの開発も同様であり、これをESの用語で言うと、専

* 日立製作所 情報システム工場 ** 日立製作所 システム開発研究所

門家の知識を知識ベース化することである。専門家の行っている問題解決能力そのものは、とうていすべて機械に移植し尽くすことはできない。しかし、機械化できるところだけでも機械化すれば、それでも十分ユーザーの役に立つ価値あるシステムができるはずだと考える。ESGUIDEでは、その機械化できる範囲をモデルとして表現したものを「概念世界」と呼ぶ。そして、その概念世界を経由して知識ベースを記述しようというわけである。

ESの開発の基本的作業を整理して図1に示す。

実世界では、人間の問題解決のノウハウが、経験と勘のようなものでコンピュータにわかることばで表現されずにある。その問題解決行動を抽象化して、専門家モデルを作成する。専門家モデルは、概念の世界での専門家問題解決行動の表現記述である。この専門家モデルをコンピュータが理解できる世界、実行できることばで記述したものがES(知識ベース)である。しかし、一足飛びにそこまで行けないので、専門家モデルをいったん形式化して、知識モデルに変換する。知識モデルは、専門家モデルをそこで利用されている知識の構造の形に重点をおいて表現したものであると言える。この知識モデルが作成された後で、構築ツールで用意されている知識表現のルールやフレームで具体化して、知識ベースに書き下す。そして、その正しさを検証する。

日立製作所が提供するES/KERNEL⁶⁾(Expert System/KERNEL)のような汎(はん)用ES構築ツールは、この「知識モデル」の枠組みを与えてくれる。そして、知識モデルから実際の知識ベースを作成する種々の手段を提供してくれる。ルールとかフレームというのは、ここで言う知識モデルの枠組みと考える。一方、ES/PROMOTE/W-DIAG⁷⁾(ES/Problem Oriented Tools for ES/Workstation-Diagnosis)のような分野別構築ツール、およびES/TOOL/W-RI⁷⁾(ES/Tool/Workstation-Rule Induction)のような知識獲得ツールは、「専門家モデル」まで用意していると考えてよい。そのため、ある専門家の問題解決行動をツールが内蔵しているモデルに近似できれば、比較的容易にシステムを開発できる。また、

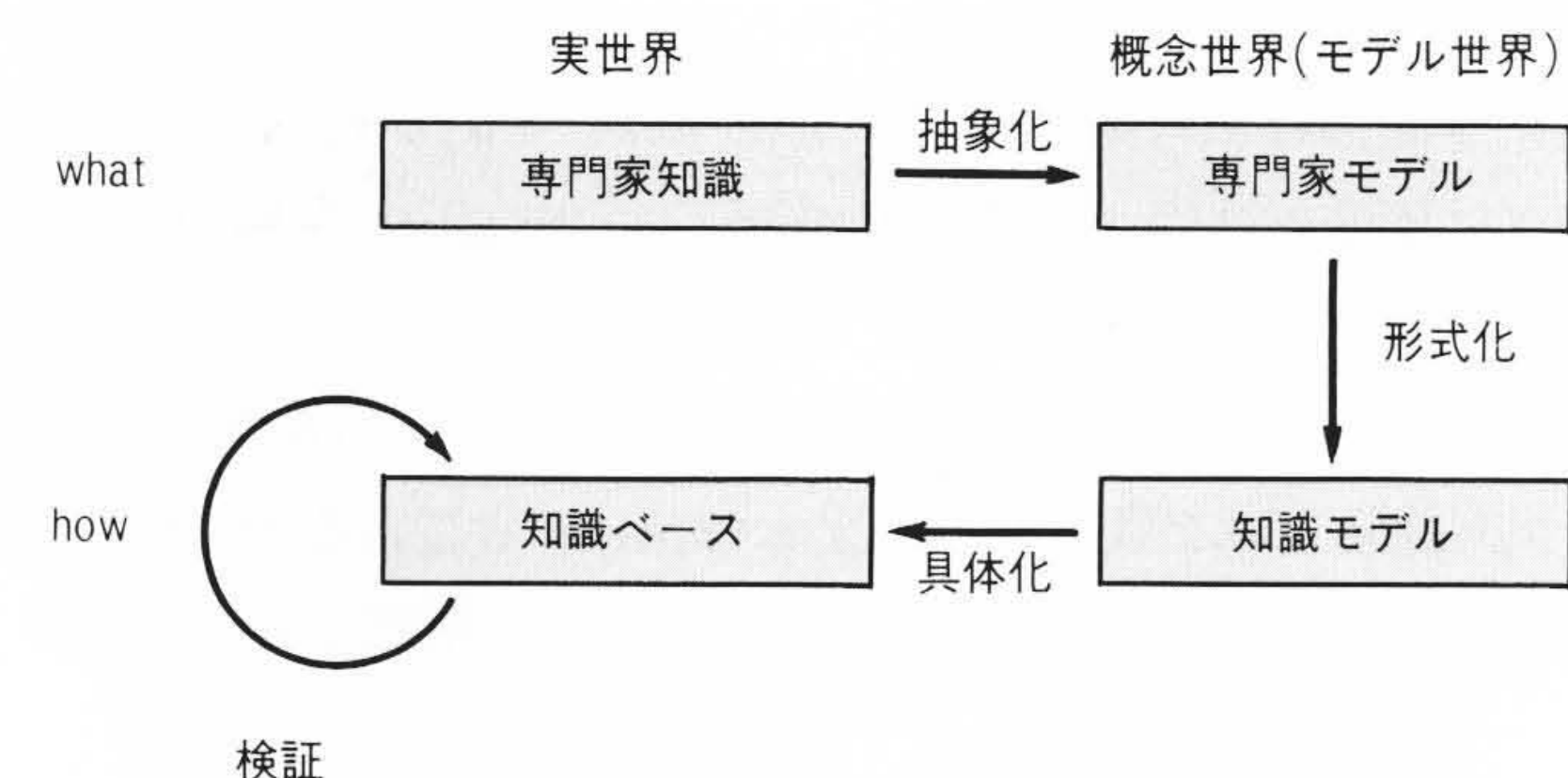


図1 システム設計の手順 実世界から、概念世界の専門家モデルと知識モデルを作成して、知識ベースを作り上げる。

知識モデルについて、汎用および分野別構築ツールで提供しているルール、フレームでは不十分な場合は、LISP(List Processor)、PROLOG(Programming in Logic)などの知識処理言語で記述することになる。

2.2 プロトタイピングアプローチの採用

ESの開発には、プロトタイピングアプローチが適していると言われている。

従来のソフトウェア開発は、ウォーターフォール形のライフサイクルが重視されていた。この方式では、ソフトウェアのライフサイクルを幾つかの明確に定義された段階に分け、おのおのの段階で行うべき作業を規定し、ステップを踏んで開発を進めていく。そして、一つの段階で規定された作業が終わらないうちは、決して次の段階に進んではならないし、後戻りすることは原則としてしてはならないと考えられている。つまり、滝とみなして全体を上流から下流に流れる滝の連なりと考えており、フェーズドアプローチとも呼ぶ。これは、おおまかにplan(計画)-do(実行)-see(評価)の手順で作業を進める。ライフサイクルを図2に示す。

しかし、ESの開発は、最初から完全な仕様が決定できない場合が多く、このようなウォーターフォールモデルの適用は難しい。

ESは、抽象化、形式化、具体化、検証という作業によって開発される。しかし、残念なことに、抽象化と形式化の境界、形式化と具体化の境界はあまり明確ではない。それらは互いに密接に関連する。一見、抽象化が開発の上流にあり、検証が下流にあると思われるかもしれないが、そのような流れとしてとらえることは難しい。実際にシステムを動かし、知識の有効性、処理性能の妥当性、各種インタフェースの妥当性を評価するうちに、そもそも「抽象化がまずかった」ということはよくある。それを失敗と考えずに、まずかったことに

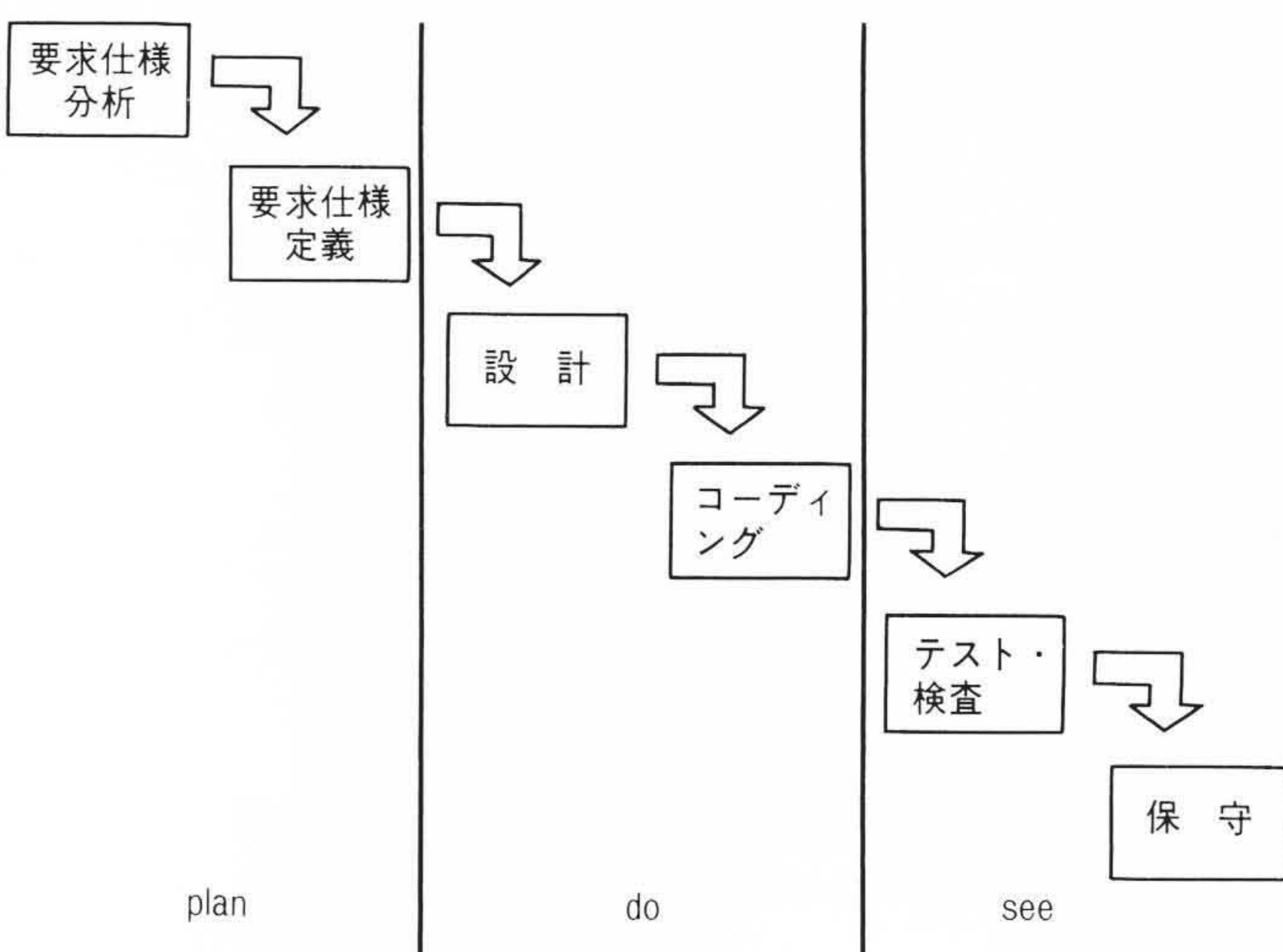


図2 従来のソフトウェアのライフサイクル(ウォーターフォールモデル) 上流から下流へと流れるように、ステップを踏んで作業を進める。

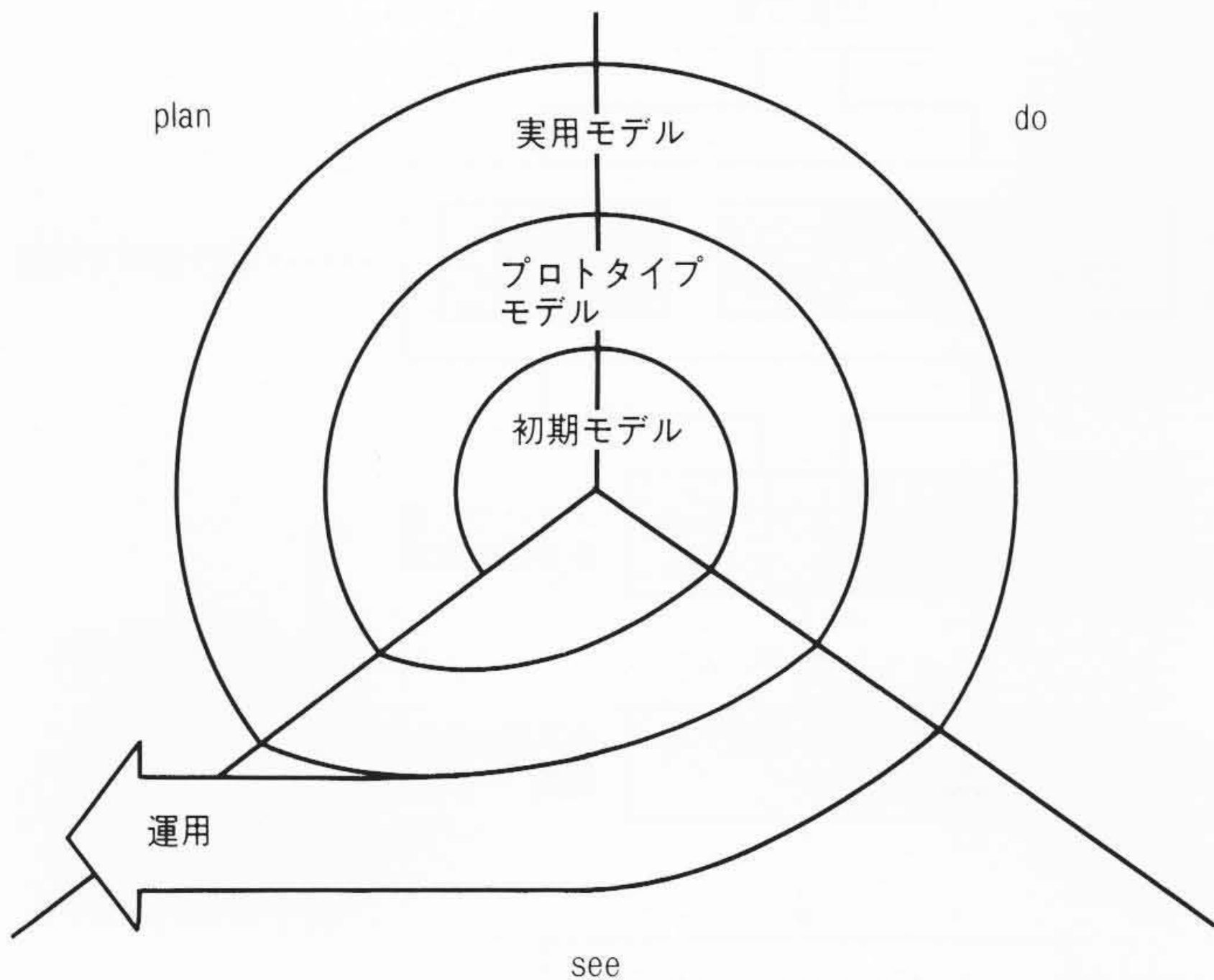


図3 ESのライフサイクル(エスカルゴモデル) 初期モデルからプロトタイプモデルの開発を繰り返しながら、実用モデルに成長させていく。

気づくことは、システム開発の一つの段階と考える。プロトタイプを作ってみて初めてわかることが大半なのである。

そこで、ESの場合は、渦巻きのように段階を踏んで開発を進めることになる。このアプローチを図3に示す。

初期モデルの開発、プロトタイプモデルの開発、実用モデルの開発へと順次進むのであるが、その過程では、何度も見直しを行い、専門家モデル、知識モデル、知識ベースが修正されていく。それは、評価と不十分な点の明確化(see)、改良のための計画(plan)、実行(do)の繰り返しでもある。このようなアプローチを「プロトタイピング」と呼ぶ。従来のフェーズドアプローチとは根本的に異なる。

ESには、まず簡単なシステムを専門家に見せないと知識が出てこないこと、あいまいなデータを扱える今までにないタイプのシステムであること、役に立つ解を出すのかわからないことなどの難しさがある。

ES/KERNELなどのツールを利用すると、知識ベースの修正が容易であり、対話画面を簡単に作ることができ、実際に動作するシステムを作ってしまうことが比較的簡単である。そこで、例えば、初期モデルでは、開発の機運を作り出す(トップの理解および専門家の協力を得る)。次に、

- (1) 有効な機能を検証する(どういう機能を持つシステムにするかの議論のたたき台にする)。
- (2) 実現の可能性を見きわめる(ほんとうに可能であることを示す)。
- (3) 処理の性能を評価する(性能がネックとなるか確認する)。
- (4) マンマシンインタフェースを評価する(使いやすいかどうかを確認する)。

ためのプロトタイプの開発を繰り返し、実用システムへと拡

張していくというようなアプローチが望まれる。

2.3 総合的な使い勝手の重視

ESに限った話ではないが、これからのシステムは使い勝手が非常に重要である。ESの場合は、特に使い勝手の良さへの期待が大きい。使い勝手の悪いシステムは、どんなにすばらしいノウハウを知識ベースに入れてあっても、エンドユーザーからは見向きもされない。さらに、運用面だけでなく、知識ベースの保守性やシステムの拡張性を考慮したインタフェースが要求される。

そのために、以下のマンマシンインタフェースの設計を重視した。

- (1) エンドユーザーインタフェース(操作性)
- (2) エキスパートインタフェース(保守性)
- (3) 既存システムインタフェース(拡張性)

2.4 ワークシート

ESの開発には、多数の関与者の一致した理解を得るための有効な手段が必要である。ESGUIDEでは、標準的なワークシートを用意した。これらは、開発者のメモ用に使用するものと、保守用ドキュメントとして使用するものがある。

3 ESGUIDEの手順

ESGUIDEは、システムの開発作業を図4に示す手順を進めるように提案している。おのこのフェーズでの作業内容を表1に示す。

3.1 構 想

ニーズ調査に基づいてESの対象業務を確定する。文献の調査や専門家へのインタビューなどを実施して、業務の内容と知識の特性を見きわめる。

特に次の2点について明確にし、業務を確定する。

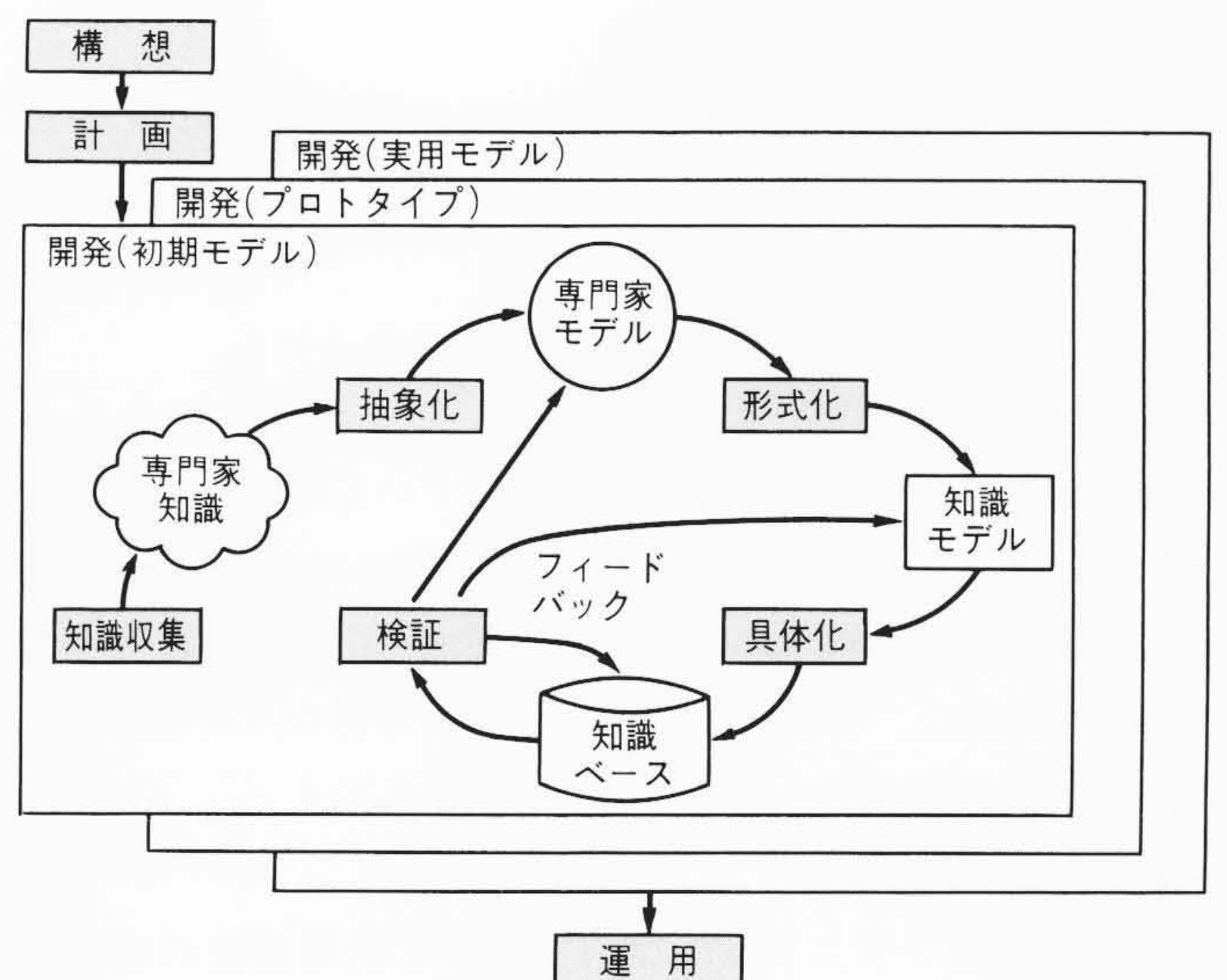


図4 ES構築手順 構想、計画、開発、運用という作業フェーズでシステムを開発する。

表1 作業内容 ESGUIDEは、構想、計画、開発、運用というフェーズにわかれて、作業内容を決めている。

作業フェーズ		概要	主な作業
構 想		対象業務の決定	・ ニーズの調査 ・ 対象業務の選定
計 画		開発計画の策定	・ 機能概要の決定
開 発	知識収集	知識の収集	・ インタビュー
	抽象化	専門家モデルの作成	・ 問題解決の枠組みの明確化 ・ 用語の定義
	形式化	知識モデルの作成	・ 知識の分類・整理 ・ システム処理フローの設定
	具体化	知識ベースの構築	・ 関連知識の記述 ・ デバッグ
	検 証	システムの検証	・ 評価・チューニング
運 用		システムの利用と保守	・ 運用体制の確立 ・ システムの利用・保守

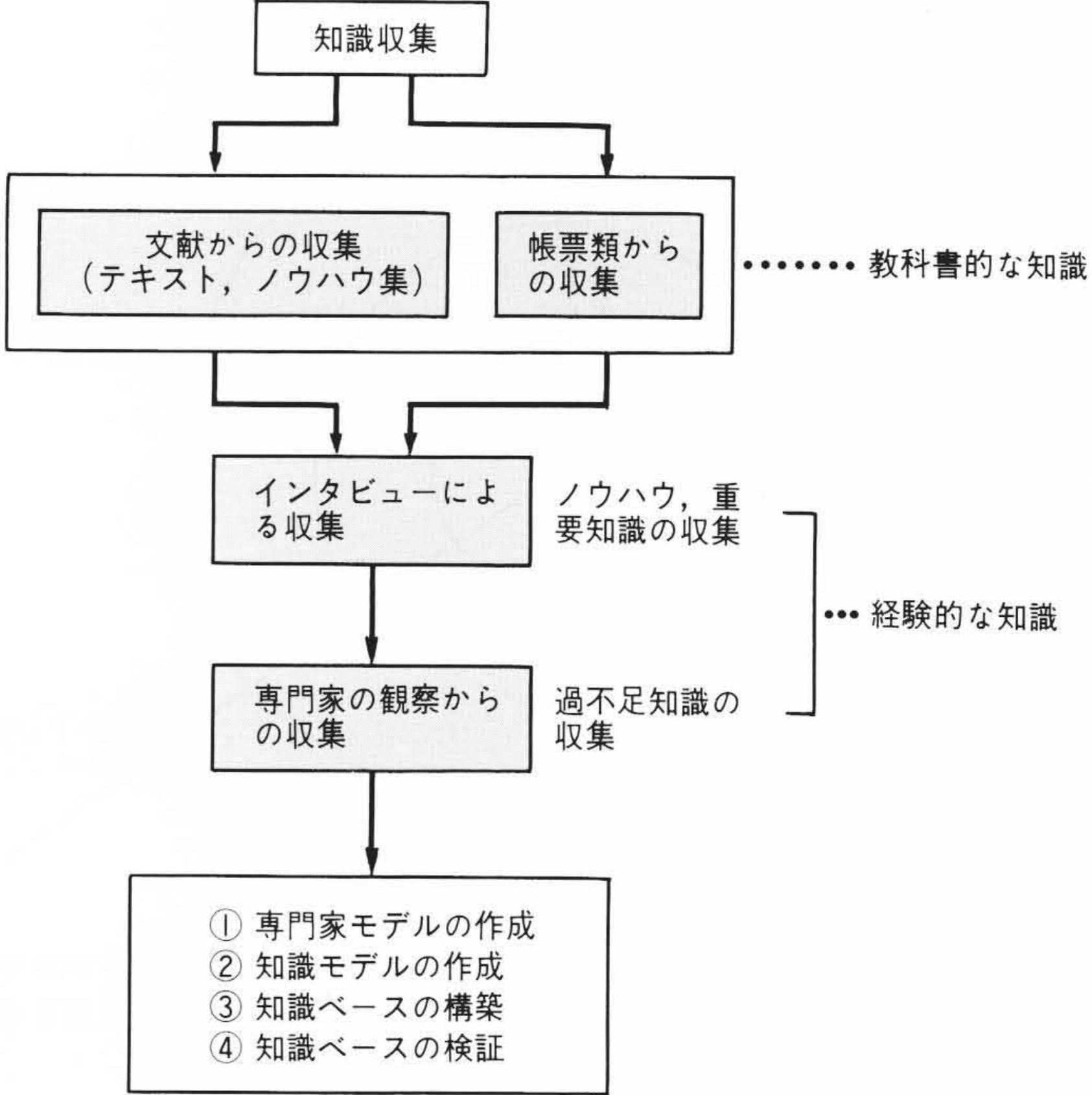


図5 知識源と収集手順 広く知られた教科書的な知識と、経験から得られた知識を抽出・収集する。

- (1) ESとして開発することの妥当性
- (2) 知識ベースの構築可能性

3.2 計 画

構想のフェーズで決定した対象業務について詳細に検討し、ESの開発作業がスムーズに行われるように開発計画を立てる。特に次の項目について明確にする。

- (1) 開発目的：最終システムは業務の効率向上、質的向上など、プロトタイプはニーズ分析、構築の可能性、性能評価など
- (2) システム化範囲：知識の範囲、システムの利用者、システムの運用形態
- (3) 機能概要：システムの評価方法、合格基準
- (4) システム構成：アプリケーションシステム、データベース、従来アルゴリズムとの接続
- (5) コストと効果の見積もり：システムの妥当性評価のための定性的・定量的効果

3.3 開 発

このフェーズはES開発作業の主要部分を占める。

3.3.1 知識収集

専門家が問題を解決する際に使用している知識を抽出・収集して、ドキュメントにまとめることがたいせつである。

知識には、問題解決の対象となる領域で広く知られた一般性のある知識と、専門家の豊富な経験から得られた知識がある。これらの知識源と収集の順序を図5に示す。

3.3.2 抽象化

システムエンジニアは、専門家による問題解決の枠組み(表2)を明らかにし、専門家モデルを作成する。「問題を解決する」ということは、「対象世界」の状態を「初期状態」から

「目標状態」に「オペレーション」を用いて移すこととみなす。

まず初めに、オペレーションと対象世界を構成する対象を決定する。例えば、レイアウト問題であれば、部屋と配置物がオブジェクトであり、オペレーションは、部屋の中にある配置物を置いたり移動させたりすることである。

「状態」とは、オブジェクトに対して与えられる状態である。レイアウト問題であれば、配置物には「未配置」の状態と「既配置」の状態があるし、部屋というオブジェクトには「配置なし」、「配置途中」、「配置完了」の状態があると考えられる。

問題そのものは、「対象世界」、「オペレーション」、「初期状態」、「目標状態」によって記述可能である。さらに、専門家による問題解決の特徴を、次の三つの要素で記述できると考える。

表2 問題解決の枠組み 基本的には、対象世界、オペレーション、初期状態、目標状態で記述する。そして、専門家が利用している問題解決の特徴を付加する。

対 象 世 界	専門家が認識する物(オブジェクト)の集まり
オペレーション	専門家が認識した対象に対する動作
初 期 状 態	対象世界の物の専門家の操作前の状態
目 標 状 態	対象世界の物の専門家の操作後の状態
仮 説	データと整合のとれる解釈
戦 略	問題解決の経緯の流れ
経 験	問題解決の過去の例

(1) 仮説を立て、代替案を作りながら評価
人間はこうすればうまくいくのではないかと考えながら、いろいろな手段を試しながら問題を解決している。そして、あらゆる可能性をしらみつぶしに試すよりも、できるだけ見込みのありそうな仮説から先に試すなど、経験に基づく知識を使っている。

(2) 許容時間内に問題を解くために戦略を使用
解を探索する効率を上げて、できるだけ短時間に解を得なければならない。専門家は、行き詰まったら後戻りしたり、問題をいくつかの部分問題に分割したり、部分解を順次求めて膨らませたり、トップダウンに解いたり、解いている途中に優先順位を変更するなどの戦略を用いる。

(3) 過去の経験を用いて効率よく解決
専門家は、過去に類似した問題があれば、そのときの解を流用する。そして、過去の失敗は繰り返さない。

専門家モデルの記述法は特に標準的なものはない。専門家の問題解決方法を記述しやすいように、自由に書けばよい。個条書きが便利なこともあれば、グラフが便利なこともある。

3.3.3 形式化

(1) 知識モデルの作成
専門家モデルを構築ツールの提供する知識表現(ルール、フレームなど)に対応づける。専門家モデルとの対応例を表3に示す。

(2) インタフェースの設計
(a) エンドユーザーインタフェース
ESの利用形態(不特定多数のユーザーが直接操作するのか、専門家が便利な道具として利用するのかなど)を整理して設計する。

(b) エキスパートインタフェース
ESは、専門家自身の手で知識を追加、修正する環境が要求されることが多い。知識ベースは、専門家による保守を対象とする部分と、対象としない部分に分ける。

(c) 既存システムインタフェース

既存のデータベースや情報処理システムと結び付けて、企業情報システムの一部として設計する。

3.3.4 具体化

知識モデルをコンピュータ上で処理できる形式にコーディングし、知識ベースを作成する。

具体化の実現方針を決定して作業を進める。知識の表現方法は複数考えられる。どのような表現にするかを左右する要素としては、冗長性をなくす、記述量を減らす、読解性を重視する、改良しやすくしておくなどあげられる。そして、これらの項目は、互いにトレードオフの関係にあるので注意する。

3.3.5 検証

- (1) 正解率：知識の正確さ、解の正答率を調べる。
- (2) 性能：対象業務の特性や利用形態を考慮して、性能が妥当であるかどうかを評価する。
- (3) 操作性：エンドユーザーが実際に利用して、マンマシンインタフェース(入力、出力)、応答時間、拡張性を評価する。
- (4) 完全性：対象としている分野の問題が、すべて解けるかどうかを確認する。

3.4 運用

実際に価値のあるESを運用維持するには、以下の点が重要である。

(1) 体制
情報処理システムの知的部分を、ESで実現することが価値ある目的である。経営者、業務部門、システム部門の役割を明確にした体制で協調してESを開発・運用することが重要である。

(2) ドキュメントの整備
定期的なテストや突発的な状況の変化を想定した知識資源管理を行う必要がある。開発のフェーズで保守を目的として作成したワークシートを使い、知識ベースを保守する。

(3) システム化範囲の拡大
企業情報システムでは、定型的ルーチン業務以外に、実績把握、意思決定、動向予測などの知的判断業務の計算機化が必要になる。生きたデータを使って、これらの問題を解決する仕掛けをESで実現することが期待できる。

そのために、従来の情報処理システムとESとの統合⁸⁾が望まれる。

4 適用事例

(1) 計画形ES
計画形ESは、ジョブショップスケジューリング、ワークスケジューリング、時間割り作成、配送計画、工程計画などがある。

計画形の問題は、与えられた制約条件を満たす組み合わせを発見する問題である。この問題が難問であるのは、すぐに

表3 専門家モデルと知識モデルの対応づけ例 ES/KERNELを利用した場合、事実形知識はフレームに、規則形知識はルールに、手続き形知識はメソッドに大別できる。

知識モデル 専門家モデル	フレーム	ルール	メソッド
対象世界	○	—	—
オペレーション	—	○	○
初期状態	○	—	—
目標状態	○	—	—
仮説	○	—	—
戦略	—	○	—
経験	○	—	—

組み合わせの爆発が起きて、実用的でなくなってしまうことである。そのため、探索を効率化するための手段を発見することが重要である。したがって、専門家モデルを作る際に、いかに上手な戦略を導き出せるかがポイントとなる。

さらに、プロトタイプの繰り返しの目的が、性能を向上させることが多い。初期モデルで数時間かかったものが、知識モデルの見直しから専門家モデルの見直しへと進めていくことにより、実用システムは数分で問題が解決できることもある。

ESGUIDEを基にして、すでにこの複雑な問題を対象としたES開発方法論⁹⁾が提案されている。これにより、実用的なESの開発^{10),11)}がいっそう盛んになった。

(2) 診断形ES

診断形ESは、機器や設備の故障診断、土木建築での工法選択、金融でのリスク診断などがある。

診断形の問題は、情報をどのような方法で入力および出力するかが重要となり、エンドユーザーインタフェースの設計がポイントとなる。イメージ、図・表、文書などを活用した事例¹²⁾が多い。また、ESGUIDEを基に、診断形ESの特徴をとらえて、診断形構築支援ツールES/PROMOTE/W-DIAGのような分野別構築ツールを提供しており、ESの開発はいっそう容易になった。

5 結 言

ESGUIDEとES/KERNELを利用することにより、従来のシステムエンジニアが特別にナレッジエンジニアリングの技術を持たなくても、ESの開発が容易になった。その一方が欠けても開発は困難な作業となる。プロトタイピングアプローチは、当初から完全な要求分析と要求仕様を決定させておく必要がないので、短期間で知識ベースの開発に着手できる。

専門家モデルと知識モデルの導入により、開発フェーズの各ステップでの目的が明らかになった。

さらに、この開発方法論を基にすると、同じ専門家モデルになる応用は、ソフトウェアの大半を再利用できる。これに着目して、分野を絞った診断形、計画形などの開発方法論を検討し、分野別構築ツールの開発も可能になりつつある。

参考文献

- 1) 特集 実用期を迎えた情報処理分野のエキスパートシステム, 日立評論, **70**, 11(昭63-11)
- 2) 安信, 外: エキスパートシステム構築標準手順“HIPACE-ES-GUIDE”, 日立評論, **70**, 11, 1105~1110(昭63-11)
- 3) エキスパートシステム構築標準手順ESGUIDE, 共通技術マニュアル, SE-362-10, (株)日立製作所(1990-8)
- 4) 花岡, 外: エキスパートシステム構築技法入門, オーム社(1990)
- 5) 片岡: ソフトウェアモデリング, 日科技連出版社(1988)
- 6) 増位, 外: エキスパートシステム技術の動向, 日立評論, **72**, 11, 1099~1104(平2-11)
- 7) 吉野, 外: エキスパートシステム構築用ミドルソフトウェアの開発, 日立評論, **72**, 11, 1119~1124(平2-11)
- 8) 花岡, 外: システムAIで開く新しい情報システム, 日立評論, **72**, 11, 1113~1118(平2-11)
- 9) 浜崎, 外: 計画・スケジューリング形エキスパートシステムへのアプローチ, 日立評論, **72**, 11, 1131~1136(平2-11)
- 10) 林, 外: 自動負荷積による納期回答エキスパートシステム, 日立評論, **72**, 11, 1179~1184(平2-11)
- 11) 福澤, 外: システムキッチン塗装工程計画エキスパートシステム, 日立評論, **72**, 11, 1185~1192(平2-11)
- 12) 松崎, 外: CPT製造不良診断エキスパートシステム, 日立評論, **72**, 11, 1193~1196(平2-11)