U.D.C. 〔681.32.06:159.95〕.002.2(083.132)

エキスパートシステム構築標準手順 “HIPACE-ESGUIDE”

Expert System Building Guidance for System Designing

安信千津子*　*Chizuko Yasunobu*
森　文彦**　*Fumihiko Mori*
花岡かほる***　*Kahoru Hanaoka*
山中止志郎***　*Toshirô Yamanaka*

エキスパートシステムを適用したいが，どのようにして開発すればよいのか分からないというケースが多いようである。エキスパートシステムの開発を促進するためには，開発方法を整備して，普及させることが早急の課題である。

日立製作所は，今までの開発経験から得たノウハウの収集，体系化を進めて，エキスパートシステム構築手順をマニュアルにまとめた。手順は，エキスパートシステムの計画，設計と並行して，知識の収集，構造化を行う。手順に沿って，作業内容，開発留意事項，ワークシートなどを提供している。昭和62年度から，社内や顧客向けのナレッジエンジニア教育に利用している。ユーザーは，マニュアルの手順に沿って，エキスパートシステムを開発できる。

1 緒　言

エキスパートシステムを適用したいが，どのようにして開発すればよいのか分からないというケースが多いようである。一方で，本号の特集にも見られるように，数多くのエキスパートシステムが開発され，開発経験が蓄積されている。

日立製作所は，開発経験から得られたノウハウを集めて体系化し，エキスパートシステム構築方法論として，顧客や社内に提供していこうとしている。既に，システム開発方法論HIPACE(Hitachi Phased Approach for High Productive Computer System's Engineering)を提案している。HIPACEの一環として，エキスパートシステム構築標準手順ESGUIDE(Expert System Building Guidance for System Designing)をまとめた。

ESGUIDEは，既にマニュアルとして製品化されている[1]。マニュアルには，開発経験に基づいた一般的なエキスパートシステムの構築手順，手順実行時の留意事項，手順に沿って作成するワークシートなどを含む。

2 ESGUIDEの概要

2.1 特　徴

ESGUIDEは以下の特徴を持つ。

(1) 開発経験に立脚

日立製作所の工場や研究所は，多くのエキスパートシステム開発に携わっている。エキスパートシステム開発経験から得られたノウハウを，収集，整理したのがESGUIDEである。実際に開発したシステムは，ビジネス分野の診断，相談，計画を行うものが多い。

ユーザーは，実際にマニュアルの手順に沿って，エキスパートシステムの開発を進めることができる。開発を進める上で問題点が生じた場合，マニュアルに記載されているノウハウを参考にして，対応策のヒントを得ることができる。

(2) ワークシートの提供

日立製作所は既に，システム開発標準手順HIPACE-SPDS(HIPACE-Standard Procedure to Develop System)[2]を開発している。HIPACE-SPDSが提案するワークシートは，広く利用されている。ESGUIDEも同様に，ワークシートベースのアプローチをとっている。

ESGUIDEは知識の記述用などの，ひな形のワークシートを提供する。エキスパートシステムを開発するナレッジエンジニアは，ひな形のワークシートをそのまま利用してもよいし，問題に合わせて修正してよい。

実際に特定分野でのエキスパートシステムを開発するときには，ナレッジエンジニアは対象分野の知識の構造を分析する。ESGUIDEのワークシートを参考にして，対象分野の知識を表現するのに適切なワークシートを設計できる。

(3) ES/KERNEL準拠

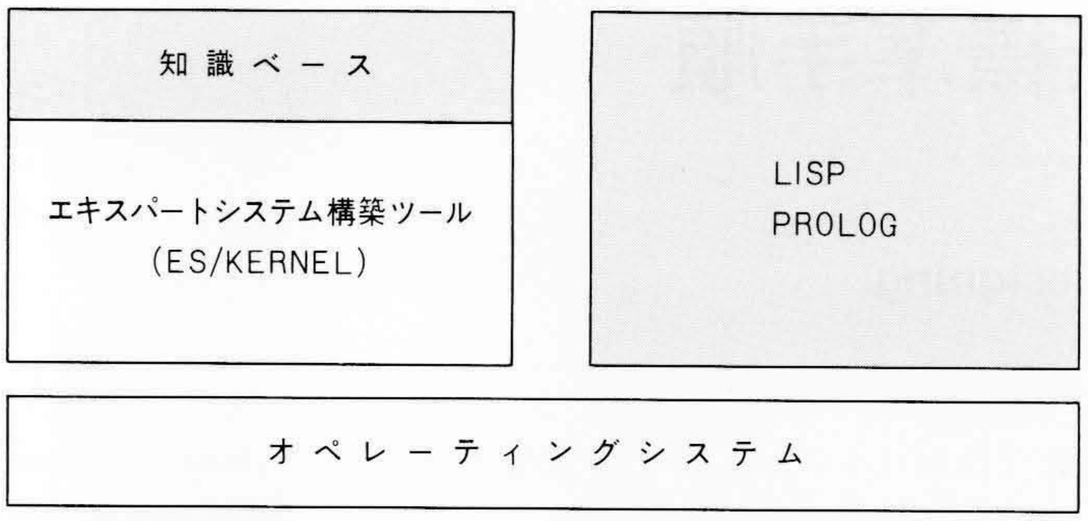
ESGUIDEの手順自体は，開発に利用するツールや言語に依存しない。手順の中の一つであるプログラムレベルの設計は，エキスパートシステム構築ツールES/KERNEL(Expert System/KERNEL)に準拠している。

ESGUIDEは一般論だけでなく，実際にES/KERNELを利用した開発の細部について記述している。ES/KERNELを利用するときに参考になる。

2.2 基本的な考え方

ESGUIDEは，次の考え方に基づいて構築標準手順をまとめ

* 日立製作所システム開発研究所　** 日立製作所システム開発研究所 工学博士　*** 日立製作所大森ソフトウェア工場



注：略語説明ほか ES/KERNEL (Expert System/KERNEL),
構築部分

図1 利用するソフトウェア ES/KERNELなどのツールや，LISP，PROLOGなどの記号処理言語を利用して開発する。

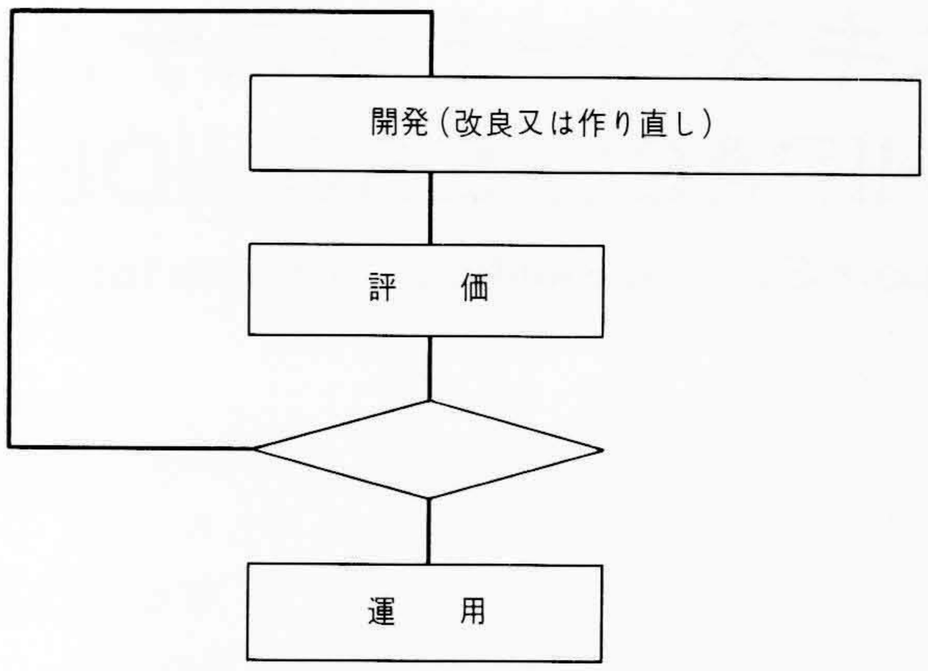


図3 プロトタイピングアプローチ 開発，評価，改良，又は作り直しを繰り返して，段階的に良いシステムにしていく。

ている。

(1) 利用するソフトウェア

ある程度対象分野の知識を収集して，必要とする知識表現と推論機能が判明した時点で，開発に利用するソフトウェアを決定できる。エキスパートシステム構築ツールを利用すると，記号処理言語を利用する場合に比較して，**図1**に示すように開発効率を上げることができる。

ESGUIDEでは，開発効率や将来の機能拡張の容易さを考えて，できるだけES/KERNELを利用することを推薦している。対象とする問題の性格に応じて，特別な知識表現や推論方式を必要とすれば，LISP，PROLOGなどの言語を利用することになる。

(2) 開発体制と保守体制

エキスパートシステムの開発・運用には，**図2**に示すように，専門家，ナレッジエンジニア，知識ベース管理者など，いろいろな役割を持つ人が参加する。

専門家を巻き込むことが，エキスパートシステムの開発には非常に重要である。専門家は，開発の最初に知識を提供して役割を終了するのではない。各フェーズでの設計レビューは，大いに専門家に参加してもらう。特に，知識のテストは専門家の意見を中心にして進める必要がある。

ナレッジエンジニアは，エキスパートシステムの設計を担当する。通常，設計したシステムの保守を，いつまでも担当することは期待できない。知識の充実・保守は，専門家だけでできるように配慮する必要がある。知識以外のシステムの保守についても，システム部門ではなく，専門家やエンドユーザー部門で，ある程度対応できるように持っていくことが望ましい。

知識ベース管理者は，知識ベースの管理に責任を持つ専門家である。知識ベースを開発するために収集，整理した知識は，専門家だけでなく企業にとっても貴重な財産である。知識を記すドキュメント，知識を保守するためのツールと体制を整備して，エキスパートシステムを運用する。

(3) プロトタイピング

最初から非常に高度な，完成されたエキスパートシステムの開発をねらうと，どこから手をつければよいのか，どのようにして開発すればよいのか分からなくなるおそれがある。いたずらに手をこまねいているよりも，可能な範囲でプロトタイプを作ることは，確実に道を一歩踏み出すことである。

プロトタイプの開発には，開発の目的を設定すること，開発したものを評価することが重要である。評価結果に応じて，**図3**に示すように，改良又は全面的作り直しを繰り返す。プロトタイピングアプローチである[3]。

もちろん無目的に開発を繰り返しても，良い方向に収束しない。プロトタイプの開発目的の例を挙げる。

(a) 開発の機運を作り出すため(トップの理解を得る，専門家の協力を得る。)

(b) 有効な機能を検証するため(どういう機能を持つシステ

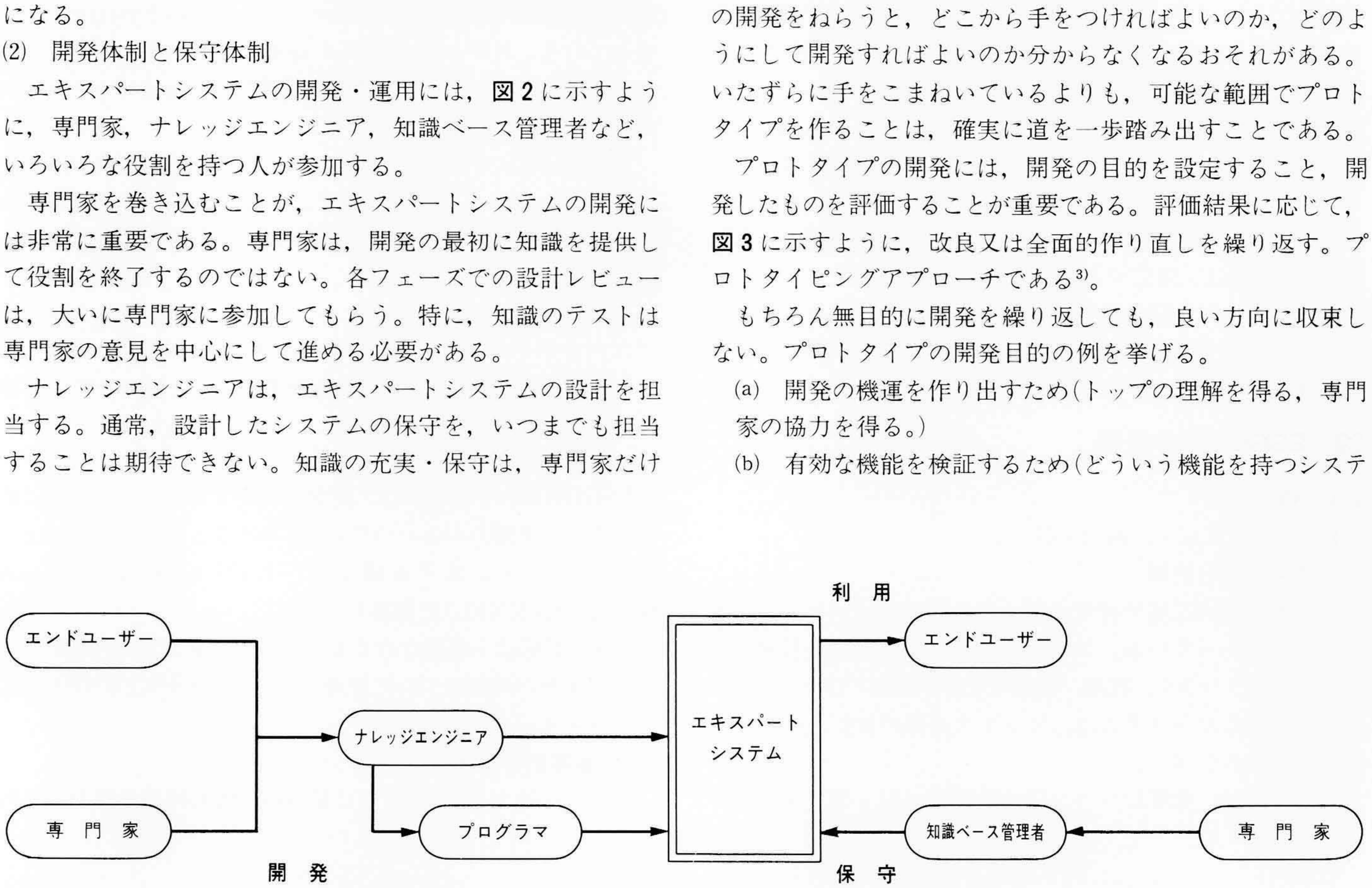


図2 開発体制と保守体制 専門家の知識やエンドユーザーのニーズに基づき，ナレッジエンジニアが中心になって開発する。

ムを作ればよいかの議論のたたき台にする。)

(c) 実現可能性を見極めるため(本当に実現可能であることを示す。)

(d) 性能を評価するため(性能がネックとなることを予想している場合は，必ず開発する。)

プロトタイピングアプローチで開発する場合，開発サイクルの繰返しパターンは，個々の開発でかなり異なる。ESGUIDEでは，1回の開発サイクルの手順に沿って記述している。

プロトタイピングアプローチで，システムを開発する場合のネックは，ユーザーインタフェースを開発するための工数である。エンドユーザーに見せる必要がある場合は，ある程度良いユーザーインタフェースを開発しないと，システムは正当に評価されないおそれがある。

ES/KERNELでは，UI(User Interface)ビルダというユーザーインタフェース構築ツールを用意している。複雑・高度なインタフェースも，プログラミングレスで短期間に開発できる(UIビルダの詳細については，本特集の関係論文を参照されたい[4])。

(4) フェーズドアプローチ

ESGUIDEでは，フェーズドアプローチ的な手順記述を採用している。確かな目的をもって開発するシステムの設計には，きちんとフェーズを分けて，段階を踏んで進めていくことが必要である。

エキスパートシステムの開発には，従来のシステム開発と比較して，「知識」と呼ばれる新しい概念が導入されている。知識に対処する方法を必要とするが，基本的に，計算機応用システムであることは変わらない。

HIPACE-SPDSのシステム分析，計画，設計などのフェーズを活用する。従来からのフェーズ(内容は多少変わっている)に，**図4**に示すように，知識を取り扱うためのフェーズを付加して，並行して進めるようにしている。

3 手順の概要

ESGUIDEの各フェーズでは，**表1**に示す作業を行う。「知識」のためのフェーズは，ESGUIDEで新たに設けたものであり，エキスパートシステム構築の主要部分を占める。

3.1 知識予備収集

文献調査，専門家面接などを実施して，問題領域の知識を

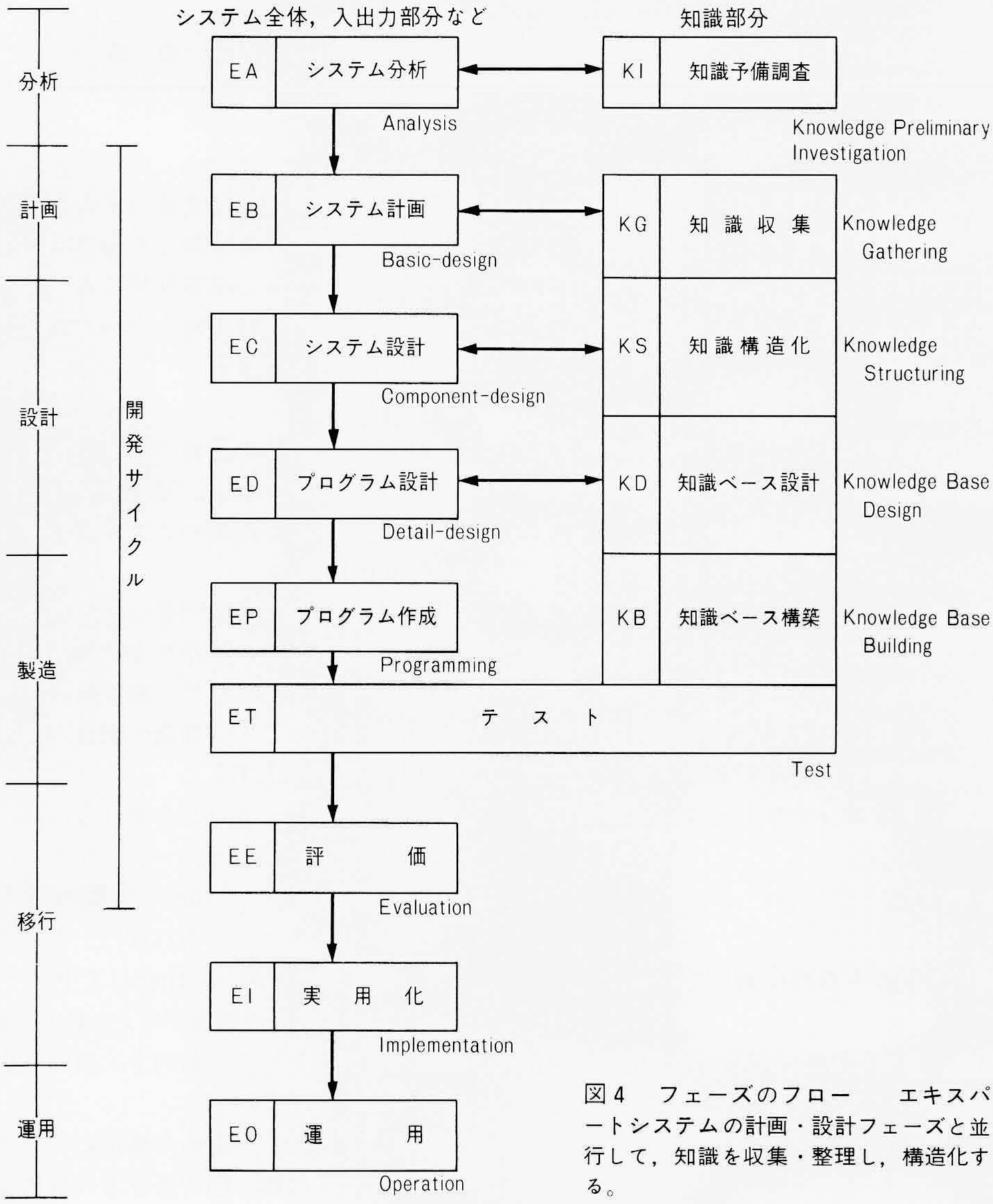


図4 フェーズのフロー エキスパートシステムの計画・設計フェーズと並行して，知識を収集・整理し，構造化する。

表1 フェーズの概要 ESGUIDEは，フェーズ別に作業内容，参加者などを定めている。

フェーズ		概要	主な作業	参加者				
				K*E	S*E	EX	EU	PG
EA	システム分析	対象業務の決定	ニーズの調査	●	●	—	○	—
			対象業務の選定					
KI	知識予備調査	知識ベース構築可能性の調査	専門家面接	●	—	○	—	—
EB	システム計画	開発範囲の確定	機能概要の決定	●	●	○	○	—
			開発計画の策定					
KG	知識収集	知識の収集と整理	文献からの収集	●	—	○	—	—
			専門家からの収集					
			事例の収集					
EC	システム設計	システム仕様の決定	機能，システムフロー設計	●	●	○	○	—
			推論方式の設計					
KS	知識構造化	知識の構造の設定	知識の記述項目の設定	●	—	○	—	—
ED	プログラム設計	プログラム仕様の決定	入出力用手続き設計	—	●	—	—	●
			制御用ルール設計					
KD	知識ベース設計	知識表現仕様の設定	専門知識ルール設計	●	—	—	—	○
			専門知識フレーム設計					
EP	プログラム作成	プログラムの開発	手続きのコーディングとテスト	—	○	—	—	●
KB	知識ベース構築	専門知識の開発	専門知識のコーディングとテスト	○	—	—	—	●
ET	テスト	知識の改良	システムのテスト	●	●	●	○	○
			専門知識の改良					
EE	評価	システムの今後の取扱いの決定	専門知識の評価	○	○	●	●	—
			全体の評価					
EI	実用化	運用へ向けての対応	機能の充実	●	●	●	●	○
			運用体制の確立					
EO	運用	システムの利用と保守	システムの利用	○	○	●	●	—
			知識の保守					

注：略語説明など
KE(ナレッジエンジニア)
SE(システムエンジニア)
EX(専門家)
EU(エンドユーザー)
PG(プログラマ)
●(主体)
○(支援)
*(SEがKEを兼任する場合もある。)

見極める。見極める際，次の2点の明確化を目的とする。

(1) エキスパートシステムとして開発することの妥当性

通常のシステムとしてではなく，エキスパートシステムの知識ベースとして開発することが適切であるかどうかチェックする。

(2) 知識ベースの構築可能性

信頼できる知識が存在するか否か，エキスパートシステム開発へ専門家の協力を得ることができるか否かについて判断する。

より具体的には，**図5**に示す点についてチェックする。

3.2 知識収集

システムが目的を遂行するのに必要な知識を，様々な知識源から収集する。収集した知識は，まず，文章，箇条書，表，図などで紙上に書き下ろす。次に，専門家の問題解決モデルを構築する。専門家の処理手順を幾つかの機能ブロックに分割し，ブロック間の関連，各ブロックの入出力，処理内容を明確にする。

主な知識源と特徴を挙げる。

(1) 教科書的な文献

知識の体系や専門用語を理解するのに適している。

(2) 専門家

実務上の知識の利用法，重要な知識と付随的な知識の区別，現実の問題に対応するためのノウハウなどを得ることができる。

(3) 観察結果

専門家が実際に業務を遂行する過程の観察結果である。専門家の処理手順，各手順で利用される知識，知識の利用方法などを把握できる。観察結果に基づいて専門家にインタビューすると，専門家が意識しないで利用しているような知識も引き出すことができる。

(4) 事例集

専門家が遂行した業務の記録である。比較的容易に大量に収集することができる。知識を検証したり，知識の漏れを防ぐのに利用する。知識のテストに必要なので必ず収集する。

3.3 知識構造化

収集した知識を，専門家の問題解決モデルのどの機能ブロックで利用するか，動作を表す規則形か，規則が参照する事実形か，などを加味して分類する。各分類に属する知識に共通の構造を見いだす。知識の構造の構成要素と各構成要素の利用方法を明確化する。

知識の構造のイメージを挙げる。

診断形であれば，

if 故障の現象 then 故障の原因

相談形のシステムであれば，

if 相談者の状況 then アドバイス

というルール形式の知識が考えられる。知識の構造とともに，知識の利用法として，前向き，後向きなどの利用する推論方式も明らかにする。

収集した知識を，定めた構造に合わせて記述するために，ワークシートの欄，図の書き方の規則などを定める。実際に

KI	知識予備調査	ワークシート	KI 1.1	知識の特性	作成		作成日		P.
					承認				

業務名			
選定項目	選定のポイント	チェック結果	推奨値
知識の変化	知識の変化頻度はどの程度か	ルール (1) 変化有(　　回／月) (2) 変化無 フレーム (1) 変化有(　　回／月) (2) 変化無	変化有 変化有
知識の定式化	知識は定式化されている業務か	(1) YES (2) NO	YES
知識の局所化	該当業務は部分的にまとまったものになっているか	(1) YES (2) NO	YES
知識の複雑性	知識はどの程度複雑か	ルール(知識) ――――― 程度（ただし，入出力のためのルールを含まない）	2桁以上
	専門家の処理時間		数分 ～数時間
	専門家の育成期間		数日 ～数年間
知識源およびその信頼性	知識源，その知識の信頼性，および専門家の参画等はどうなるか	知識源 (1) (2) 上記知識源の信頼性 (1) 有 (2) 無 (3) 不明 専門家の参画 (1) 可能 (2) 不可能	マニュアル，専門家 有 可能

図5　専門知識のチェックポイント　　エキスパートシステム開発の対象分野を選定する段階で，対象分野の専門知識が適切であるか否か，チェックする。

知識を記述してみて，不足を補ったり整合性をチェックしたりする。

3.4　知識ベース設計

収集，整理した知識を，利用するツールや言語の知識表現として実現する方法を設計する。設計するには，知識としての記述の自然さ，同時に変更・保守される可能性のある知識のまとまりの良さ，推論実行時の性能などを考慮する。

通常，次の2段階に分けて考える。

(1)　理想を追求して，知識を自然にすなおに記述する。

(2)　現実に妥協して，ツールの機能の制限や性能を考慮する。

構造化して記述した知識を，機械的にルール，フレームなどの知識表現に変換できるようにするとよい。ユーザーは，知識をワークシートなどに記述しさえすれば，推論で利用することができるようになる。

3.5　知識ベース構築

構造化して記述した知識を，利用するソフトウェアで推論可能な知識としてコーディングする。具体的には，知識構造化フェーズでワークシートに整理した知識を，知識ベース設計フェーズで決めた規則に基づいて変換する。

3.6　テスト

いろいろな事例で推論を実行して，満足のいく結果が得られるか否かチェックする。満足のいく結果が得られなかった場合は，推論のどの段階で問題が生じているかを調べて知識を修正する。

テストを漏れなく効率的に行うため，テスト事例，テスト結果，問題が生じている事例の推論過程解析結果などを記述するワークシートを利用する。

テストに用いる事例は，次の順番に利用する。

(1)　知識収集に利用した事例

知識を収集するのに利用した事例であるので，良い結果を得るのが当然である。悪い結果を得た場合，原因を調査して考え違いがあれば，知識の構造化や設計を見直す必要が生じる可能性がある。

(2)　まだ利用されていない知識を必要とする事例

(1)のテストだけでは，一度も利用されていない知識が残っている可能性がある。利用されていない知識を利用する事例を作成する。

(3)　特殊な事例

特殊な状況を想定して，テスト事例を作成する。特殊事例を処理するための知識を，追加する必要が生じる場合も多い。

4　適用例

架空の例であるが，自動車選択エキスパートシステムの知識整理に，ESGUIDEを適用する。自動車選択エキスパートシステムは，自動車のショールームなどに設置されて，自動車の購入を考えている人が利用する。利用者の要求に合う車種を推薦する。

4.1　専門家の問題解決モデル

専門家の問題解決モデルは，**図6**に示すように二つの機能ブロックに分かれている。一番目の「項目別の検討」機能は，予算，定員などの利用者の要求項目に関して，各車種が利用者の要求を満足するか否かを判定する。二番目の「総合判断」機能は，項目別の満足状況から総合的に判断して，各車種を推薦するか否かを決める。

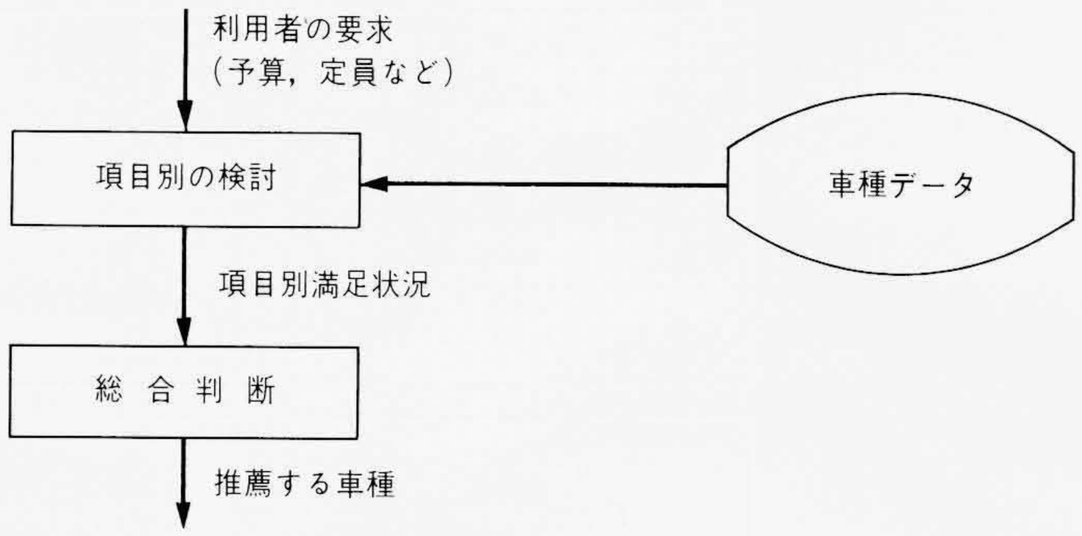


図6　自動車選択アドバイザの問題解決モデル　　専門家は，まず予算，定員など4要求項目について検討し，次に各検討結果に基づいて総合判断を行う。

4.2　知識の構造化

知識は，**表2**に示すような構造で記述している。「項目別の検討」では，事実形知識として車種のデータ，規則形知識として検討項目別の判断基準に関する知識を利用する。「総合判

```
(プライミニスタ
    class    自家用車
    価 格     500
    定 員       5
    排気量   5000
    タイプ   {デラックス}
)
```

他の車種も同様

```
(価格ルール
もし   (要求 の @価格 を ?予算とする)
       (?自動車 の @class が 自家用車 であり
                @価格 が ?予算 以下である)
ならば ・〔?ある自動車〕 が価格からみて適当.
(
```

他の検討項目のルールも同様

```
(価格肯定ルール
もし     〔?ある自動車〕 が価格からみて適当.
ならば   〔?ある自動車〕 を推薦する 0.5.
)
```

他の検討項目のルールも同様

```
(推薦ルール
もし
  〔?ある自動車〕を推薦する ＞=0.8.
ならば
  (send system print (" 〔推論結果〕 %s を 推薦しますYn, ?ある自動車))
)
```

図7 自動車選択の知識表現 知識ベースは，車種データを表現するフレーム，価格について検討するルール，推薦の判断を行うルールなどから成る。

表2 自動車選択の構造化した知識 知識を，車種，検討項目及び総合判断に関するものの3種類に分類して，構造化して記述している。

(a) 車種に関する知識(事実形)

No.	車 名	価格(万円)	定員(人)	排気量(cc)	タ イ プ
1	プライミニスタ	500	5	5,000	デラックス
2	セ ー ラ	300	5	3,000	デラックス
3	フェアボーイ	350	2	3,000	スポーツ
4	シーライン	200	5	2,000	ファミリー，スポーツ
5	レジスター	150	4	1,500	ファミリー

(b) 検討項目(規則形)

No.	項 目	適切と判断する車種
1	価 格	価格が購入資金以下
2	使用目的	タイプが要求タイプと一致する*。
3	定 員	定員が要求された乗車人員以上
4	排 気 量	排気量が要求された排気量以上

注：* 複数のタイプがある車種は，要求のタイプが車種のタイプの一つであれば適当である。

(c) 総合判断(規則形)

検 討 項 目	確 信 度
価 格	0.5
使 用 目 的	0.5
定 員	0.5
排 気 量	0.5

注1：確信度の合計が0.8を超えると推薦する。3項目が適切の場合0.875，2項目が適切の場合0.75となる。

2：各検討項目の確信度は最初はすべて等しいと仮定し，テストしながら調整する。合格基準も同様である。調整しやすいようにルールを設計する。

断」では，推薦するか否かを判断する規則形の知識を利用する。

4.3 知識のコーディング例

知識は，**図7**に示すようなフレームやルールの形式でコーディングしている。車種のデータの実現するフレーム，検討項目別の満足状況について判断するルール，総合判断のためのルールなどである。

5 結 言

ESGUIDEは，社内で適用して検証を進めている。昭和62年度から社内や顧客向けのナレッジエンジニア教育に利用されている。

更に，ナレッジエンジニアになろうとするのではないユーザーの便宜を図るため，入門編のマニュアルや，より具体的な内容の分野別マニュアルを作成している。

参考文献

1) システム開発標準手順SPDS 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，(株)日立製作所(昭62-10)

2) システム開発標準手順SPDS概説編，共通技術マニュアル，SE-160，(株)日立製作所(昭60-5)

3) J.Martin：Application Development without Programmers，Prentice-Hall(1982)

4) 増石，外：ES/KERNEL/Wのユーザーインタフェース構築ツール「UIビルダ」，日立評論，**70**，11，1100～1104(昭63-11)