U.D.C. 681.32.06.002:〔002.5:681.32.06(083.75)〕
:〔519.767.6:809.56〕

システム仕様書の再利用によるソフトウェアの開発技法（ICAS-REUSE）

Software Development by Reusing System Development Specifications

千吉良英毅* *Eiki Chigira*
永松祐嗣* *Yuji Nagamatsu*
小林正和* *Masakazu Kobayashi*

新しくソフトウェアを開発する場合の生産性向上を目的としたソフトウェア生産支援ツールが，従来開発されてきた。この場合は，類似ソフトウェアでも新規に開発することが多く，生産性に限界があった。ソフトウェアを部品に分解できれば，部品レベルで再利用が可能となる。従来はプログラムを部品化する方法がよく採られてきた。それよりも高い生産性をねらって，より上流工程の部品化を意図して，システム仕様書の再利用を行った。既存システムを最大限再利用するためには，開発対象システムに類似した仕様書の検索が重要な技術となる。ICAS-REUSEは，要求文解読のための日本語処理機能，要求文の欠落情報を補足するための業務上の概念依存関係を用いた推論機能，類似仕様書の検索機能などを備えている。

1 緒　　言

計算機ソフトウェアの需要は増大し，更にソフトウェア機能は高度化，複雑化，大規模化している。

ソフトウェアの生産性，信頼性に対処するためには，生産支援ツールを用い信頼性の高いソフトウェアを常に新たに開発してゆく方式と，既存ソフトウェアを再利用する方式がある。

ソフトウェアを常に新しく開発する方式は，標準化技法やツールなどを用い信頼性の高いソフトウェアを効率よく作成することを主眼としたものである。一方，ソフトウェア再利用方式は，再利用できるソフトウェアを最大限流用し，新たに開発するソフトウェアの量を最小限にするものであり，ソフトウェアの生産性と信頼性を高める上で重要な一方式である。

よく使われるプログラムを部品化し，部品の再利用を支援するツールは既に実用化されている[1),2)]。プログラム部品の修正が生じないようにソフトウェアを設計するためには，設計者がプログラム部品に習熟していることが必要である。

既存ソフトウェアの仕様書を再利用する技術(ICAS-REUSE：Integrated Computer Aided Software Engineering System-Reusable Software Engineering)では，新たに開発しようとしているシステムに似ている仕様を備えたシステムの仕様書を，検索することが重要な課題である。これには要求を解読することと，要求の漏れを意味的に補足することが必要である。ユーザーの言葉(業務の言葉)で記述された要求を解読するために，情報処理用格文法辞書を用意した。要求の漏れを推論するために，業務上の概念依存関係をあらかじめ登録することとした。

ICAS-REUSEは，要求がユーザーの言葉で定義できること，最大限再利用できる既存システムの仕様書が得られることなどの特徴があり，業務用語で書かれた承認仕様書の迅速な作成や類似仕様書の再利用による効率的なシステム設計が実現できる。

ソフトウェアの開発過程は，**表1**に示すように6段階に分けることが多い。ICAS-REUSEはシステム設計以降を支援する技術である。

2 ICAS-REUSEの設計思想

システム設計では，ユーザーの要求を正確に把握し，要求を過不足のないシステム機能で満たすことが目標となる。

システム設計では，おおむね次のような作業手順を踏む。

(1) ユーザーの要求を，フリーハンドで描いた絵，図，表，情報の流れ図，文章などを用い，各業務ごとに記述(非定型的記述)する。

(2) システム設計者は，ユーザーの要求をシステムの仕様として見直し，冗長な部分，欠落した部分を要求元(ユーザー)に確認しながら補い，ユーザーと合意のとれたシステムの仕様書を作成する。

(3) 作成したシステム仕様書に基づいて詳細設計を行う。

このような手順を踏まえ，設計工程の生産性・信頼性を向上させるために次のような基本方針を設定した。

* 日立製作所システム開発研究所

表1 ソフトウェアライフサイクルの各フェースの作業概要
REUSEは，システム設計フェーズ以降の支援を目標とする技術である。

	ライフサイクルフェーズ	作業の目標
1.	システム分析計画	●システム化の目的と解決手段の明確化 ●プロジェクトの到達点及び目標の設定
2.	システム設計	●ユーザー要求の過不足のない仕様化 ●ソフトウェア構成要素の決定
3.	ソフトウェア設計	●ソフトウェア機能要求を実現するソフトウェア方式の決定 ●設計の経済的な実現（モジュール化，モジュール間インタフェース，モジュール内論理）
4.	ソフトウェア製造	●ソフトウェア設計仕様の計算機言語への翻訳(コーディング) ●計算機言語への翻訳誤りの除去(デバッグ)
5.	システムテスト	●作成されたシステム全体の要求との合致を検査(機能，品質)
6.	運用・保守	●完成システムの使用と誤りの補修 ●使用環境変化に応じた修正・改良

(1) あいまいな要求から，定型的なシステム仕様書の記述へ変換する過程を支援する機能を導入する(要求の翻訳)。

変換過程の支援機能は，次のようなサブ機能で構成する。

(a) あいまいな要求を定型的な記述の仕様書に変換する機能

(b) 変換・生成したシステム仕様書から冗長性を削除する機能

(c) 変換・生成したシステム仕様書から欠落している情報を付加する機能

(2) 既開発システムの仕様の中から要求機能を含む仕様を抽出し，システム詳細設計を支援する機能を導入する(設計情報の再利用)。

以上の基本方針と機能概要(**図1**)に基づいたICAS-REUSEの操作と機能のフローを**図2**に示す。

3 ICAS-REUSEで実現した基本機能

3.1 要求をシステム仕様に翻訳する機能

翻訳機能は，要求を解読しシステム仕様書に変換する機能と，システム仕様書としての冗長性を削除する機能，欠落情報を付加する機能から構成される。

3.1.1 要求記述手段及びシステム仕様書への変換機能

要求を記述する手段は多数ある。記述する規則が厳密なものほど仕様としての完備性が保証される。しかし，利用者にとっては記述規則を忠実に守らねばならないため，使いにくい場合もある。

REUSEでは，要求記述手段として日本語文を用いた。利用者の利便性を考慮し，使用する日本語文に次のような記述内容の制約と構文上の制約を設けた。

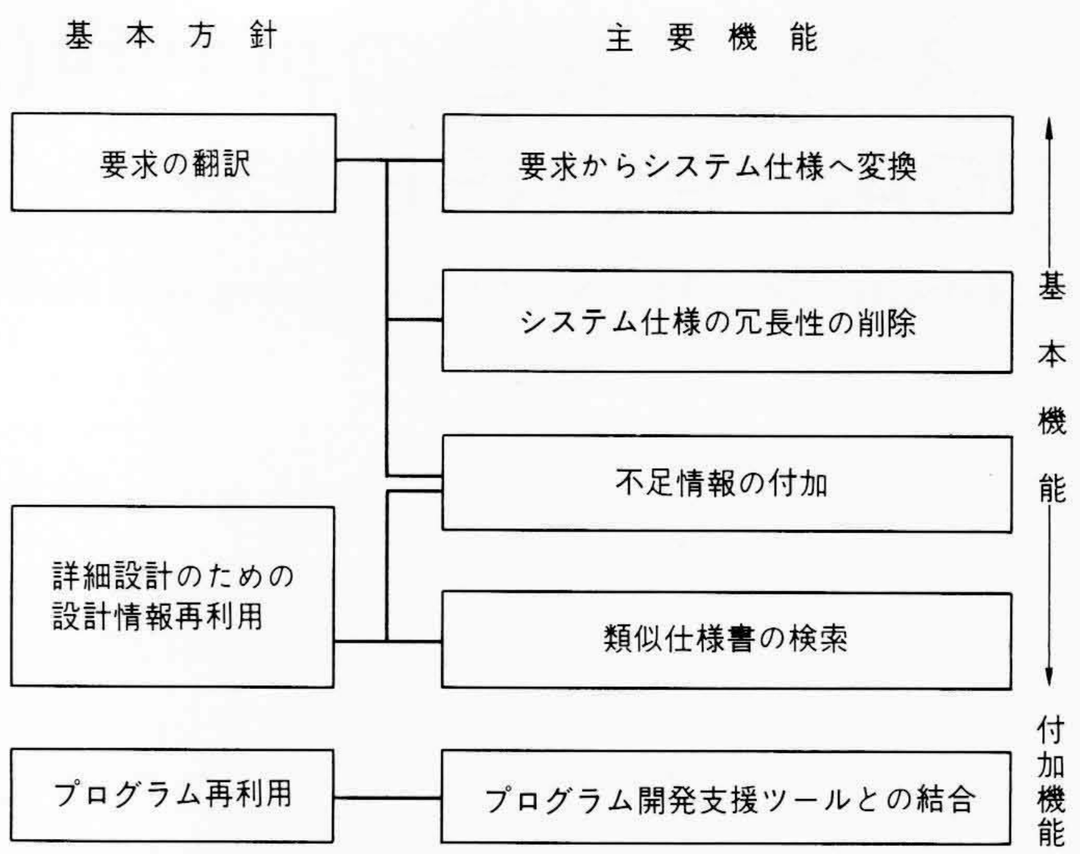

図1 REUSEの設計方針 ソフトウェアの再利用を，システム開発の早期から実施できるように機能を設定した。

(1) 記述内容の制約

(a) システムの機能を記述すること。

(b) システムの機能をトップダウン的に記述すること。

(2) 記述構文の制約

(a) 単文又は単文を結合した重文を原則とする。

(b) システム機能の起動条件を記述する場合，定められた用語(14種)による複文を用いることができる。

(c) 前後する二つの文が互いの機能の出力，入力の関係で意味的に結ばれている場合，指示代名詞を用いた係り受けのある文章を用いることができる。

以上のような制約のもとで，要求を実際に日本語文で記述すると次のような問題が生じた。

(a) 業務固有の用語が多用されること。

(b) 業務上常識に当たる内容は記述されないことが多い。

(c) 業務のデータに関する記述漏れが多い。

(d) 業務処理の上位・下位に関する記述漏れが多い。

これらの問題は，日本語文を使用して要求を記述したことに起因すると思われる。REUSEでは，これらの問題に対処するため，以下の日本語辞書を用意した。

(a) 日常用語辞書※1)

(b) 業務用語辞書

業務用語を認識するために利用する。業務用語辞書は用語ごとに用語間の部分を示す語と全体を示す語との関係，同意語などを登録する。

(c) 格文法辞書

格文法解析を実施するために利用する。

システムの機能を表わす動詞の格を定めた(10種類の格を設

※1) REUSEは，日常用語に関し，日英機械翻訳システムHICATS/JE (Hitachi Computer Aided Translation System/Japanese to English)の用語辞書を利用した。

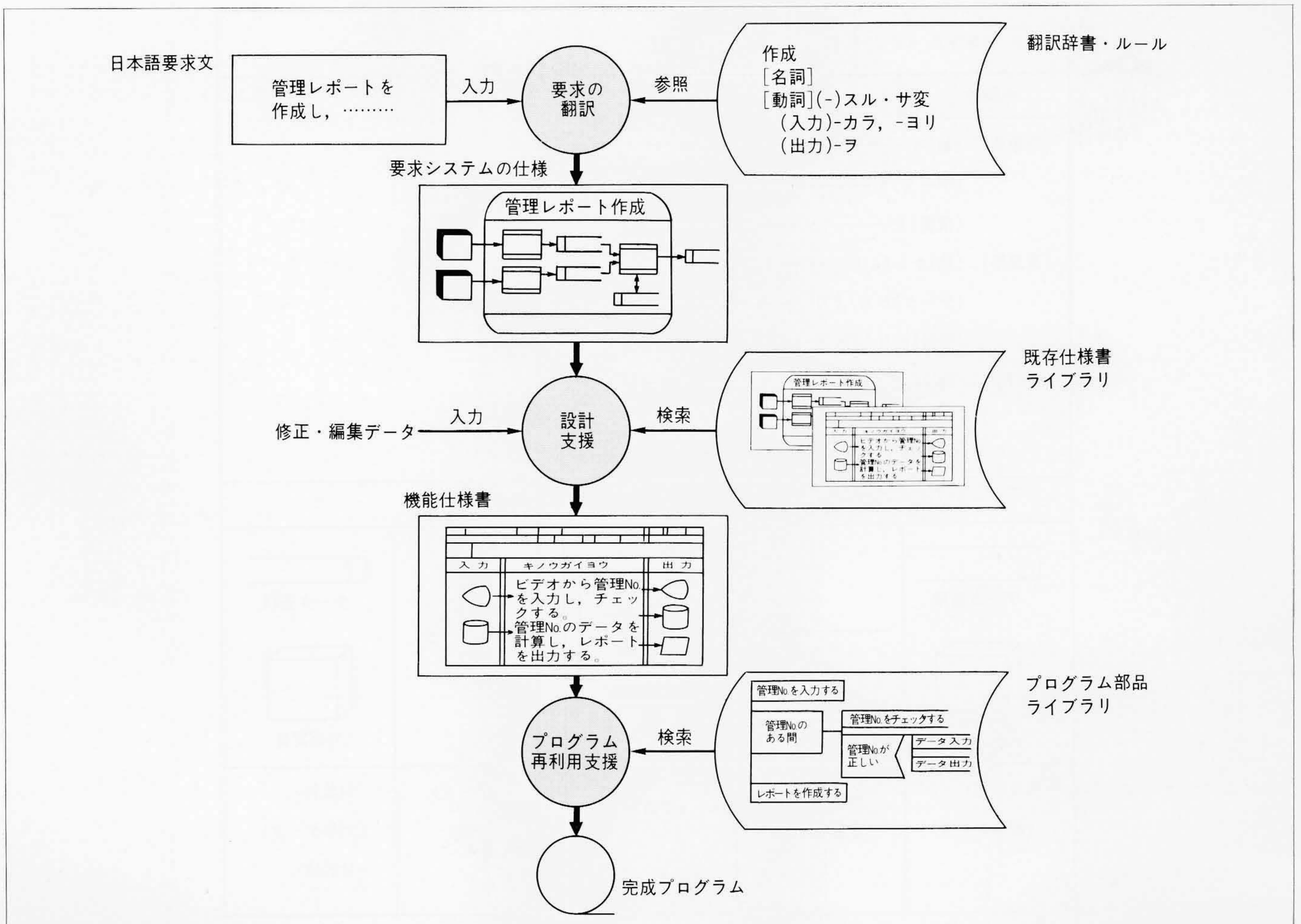

図2　REUSEの機能の流れ概要　　要求を翻訳してシステム仕様を作る。類似仕様を検索修正し，ソフトウェア仕様を完成する。プログラム再利用ツールにつなぐ。

定)。各格に，物，データ，情報，方法及び条件という5種のデータフロー変換属性を設定した。この属性のデータは，業務用語辞書に登録する。

REUSEは要求を記述した日本語文(要求文)を，システム機能とデータとの関係図(データフロー図)へ変換する。変換時データフロー変換属性データと格文法辞書を利用する(**図3**)。

3.1.2　システム仕様書の冗長性を削除する機能

システムの仕様を記述する方法も多種多様であるが，記述しやすく読みやすいデータフロー図を利用する。

要求文は，複数の文章で構成される。変換機能により各文ごとに1個のデータフロー図を生成し，これらを統合してシステム仕様書を作成する。

設計者は，数個のデータフロー図を指定して統合することも可能である。また，データフロー図の構成要素(システム機能，データ)を指定して統合することも認められる。

本機能により設計者は機械的にデータフロー図を統合し，冗長性を削除できること，及び望むとおりの編集を行うことができる。

3.1.3　要求文に書かれていない情報の付加機能

日本語文章の持つ冗長性を考慮すると，要求文にすべての要求が書かれているという保証はない。むしろ記述漏れがある場合が多い。要求文の解析結果に基づいて意味上の補足を行い，隠れた要求を推論する必要がある。

意味付加を目的とした推論を実行するために，一定の規範に基づいて構築した知識体系が重要である。REUSEは業務固有の概念，物，ノウハウの定義と各々の関係を，フレームと意味ネットワークとを用いて知識化(概念依存関係)し，要求文からの欠落情報を推論(意味付加推論)する機能を備えている。

システム仕様やソフトウェア仕様は，機能中心にまとめる場合が多いため，ある業務の処理手順やノウハウを，処理や機能を中心に体系化することもよくとられる方法である。しかし，処理や機能はまとめる人によってとらえ方が異なることがあり，人による差異のない体系を確立することは困難なことが多い。これに対し，業務で実際に使用している物は特定できるため差は生じにくい。したがって，業務で実際に使われている物を中心に，付随する概念をも含めて体系化を図った。

REUSEの適用業務に関する知識体系は，次のように分類して構築した。

対象となる物が存在する場合，対象物の種類により次のように分けて考えることができる。

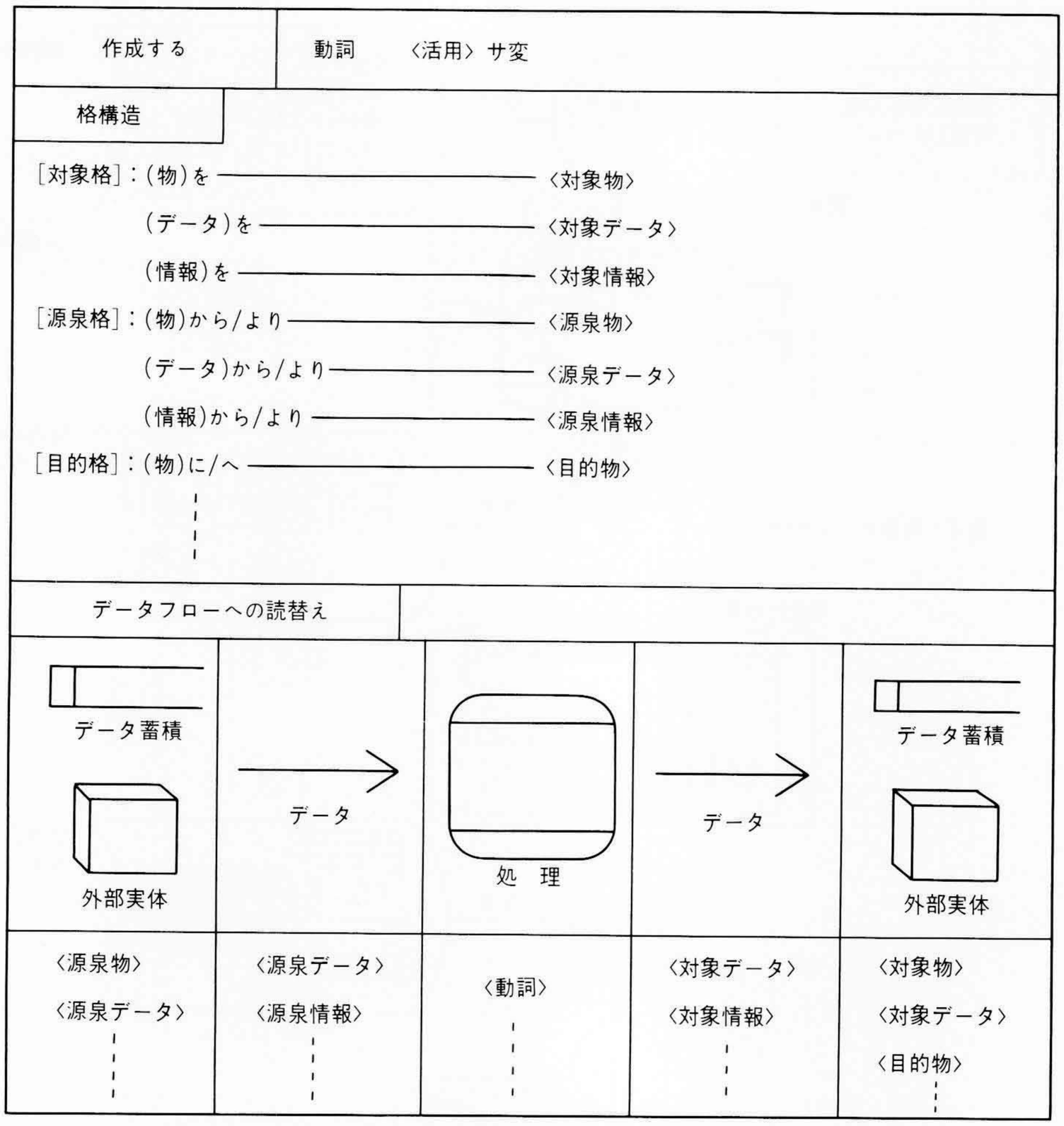

図3 格文法辞書 情報処理用の格パターンとデータフローへの読替えのパターンを備えている。

(1) 人及び組織に関するもの
例えば，販売員，仕入係，営業部など
(2) 金銭に関するもの
例えば，一万円札，クレジットカードなど
(3) 物に関するもの
例えば，衣料品，電子会計機など

概念の場合は「情報」，「データ」，「時間・期間」，「単位」，「数・量」，「比率・割合」，「条件」などに分けて考えることができる。

このような対象物及び概念に関し対象指向モデルを適用し，対象物又は概念と業務処理を組にして一つの知識の単位とした。更に，対象物又は概念を分割あるいは抽象化し，知識を構造化した。これらの知識は，分割の場合は“part of”※2)，抽象化の場合は抽象化の尺度名を付け，“is a”※3)の名称で関係づけた(**図4**)。

REUSEは以上のような知識と推論を実行するルールを用い，要求文に記述されていない隠れた要求を設計者と対話しながら抽出し，データフロー図へ変換する。

3.2 仕様情報の修正を支援する機能

既存仕様書を再利用し，システム設計・ソフトウェア設計を支援するためには，既存仕様書の検索機能と仕様書の編集機能が必要である。

REUSEでは仕様書ライブラリに既存仕様書を登録し，これを検索し，ワークステーション上で修正編集を行う方式を採用している。

要求文を翻訳し，設計者との対話によって作成したデータフロー図と似ている既存システムのデータフロー図を検索し表示する。この際，記述されている情報で一致するものが多いほど，より似ている仕様書と考える。データフロー図の表示に当たっては，完全に一致している情報，修正を必要とする情報，全く一致しない情報を色分け表示する。設計者は，修正量が少なく複雑な修正を施さなくて済むものを選択で

※2) “part of”は，意味ネットワーク上の関係の一種で，例えば，商品コードは商品マスタファイルの一部分(“part of”)であるというように，ある対象と別の対象との関係で特に全体と部分の関係を表す。

※3) “is a”は，意味ネットワーク上の関係の一種で，例えば，商品台帳は台帳の一種(“is a”)であるというように，ある対象と別の対象との関係で特に抽象と具体の関係を表す。

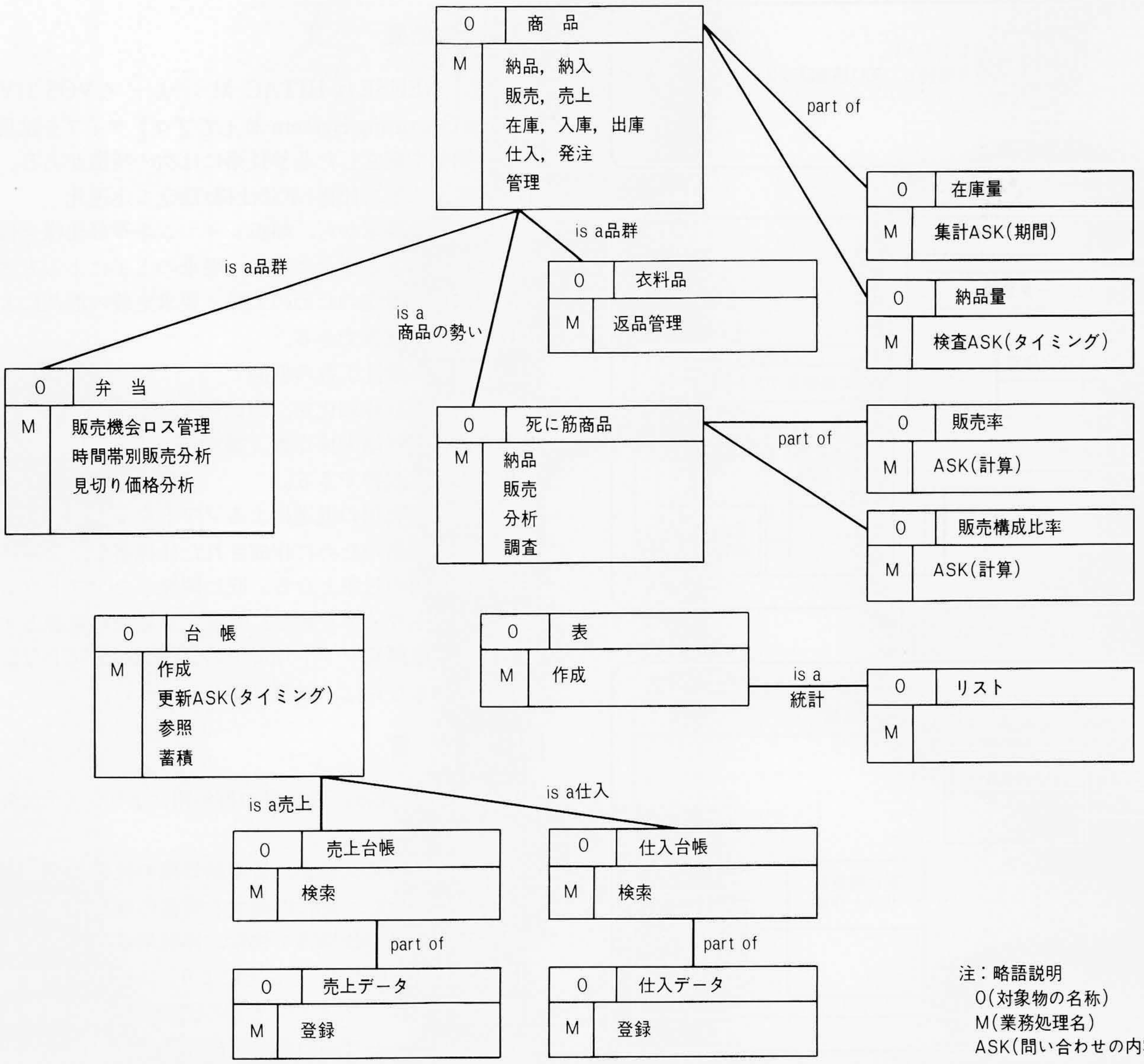

図4　物を中心とした商品関係の概念依存関係　　商品に関連する基本的な概念や「もの」と相互の関係を記述する。

きる。

再利用するシステムのデータフロー図を決定すると，関連する仕様書はすべて検索可能となる。したがって，設計者が必要とする仕様書を検索し，再利用エディタで修正編集し最終的にソフトウェア仕様書を完成できる。

4　適 用 例

要求文の解読から既存仕様書を検索するためのデータフロー図の作成までの手順を，「死に筋商品リスト出力」の例を用いて説明する（**図5**）。

要求文「売上データを登録し，死に筋商品リストを作成する」を入力する〔**図5**(a)〕。日本語解析を行い，「売上データ登録」と「死に筋商品リスト作成」の2種の未完成データフロー図を生成する〔**図5**(b)〕。未完成のまま両者を統合する〔**図5**(c)〕。売上データを対象物の中から探し（**図4**），「売上データを登録し，売上台帳を作成する」を描出する（作成するには上位の台帳の作成業務からインヘリット※4)する）。死に筋商品を対象物の中から探し（**図4**），更にリストを探し（**図4**），両者を合成した新しい知識単位「死に筋商品リスト」を「リスト」と“is a”（リンク名は死に筋商品リスト），「死に筋商品」と“part of”で結ぶ。これにより「死に筋商品を分析し，死に筋商品リストを作成する」を描出する。次に「死に筋商品」と「売上台帳」を合成し「売上台帳を検索し，死に筋商品を分析する」を得る。**図4**の範囲内の知識では，品群の選定が残っており，これは問い合わせとなる〔**図5**(d)〕。要求文の記述が重文でなされ，主文が「死に筋分析リストを作成する」であったので，本システムの名称を「衣料品死に筋分析」として**図5**(e)が出力される。この図は，設計者が「販売員，仕入係，販売統計」を編集入力したものである。

本事例で述べた推論時に生成した知識単位は，生成頻度とともに一定期間保存し，必要に応じて知識体系の中に組み込むこともできる。

※4)　“is a”で結合された対象物が，上位対象物の業務処理を利用する機能をインヘリットと呼ぶ。

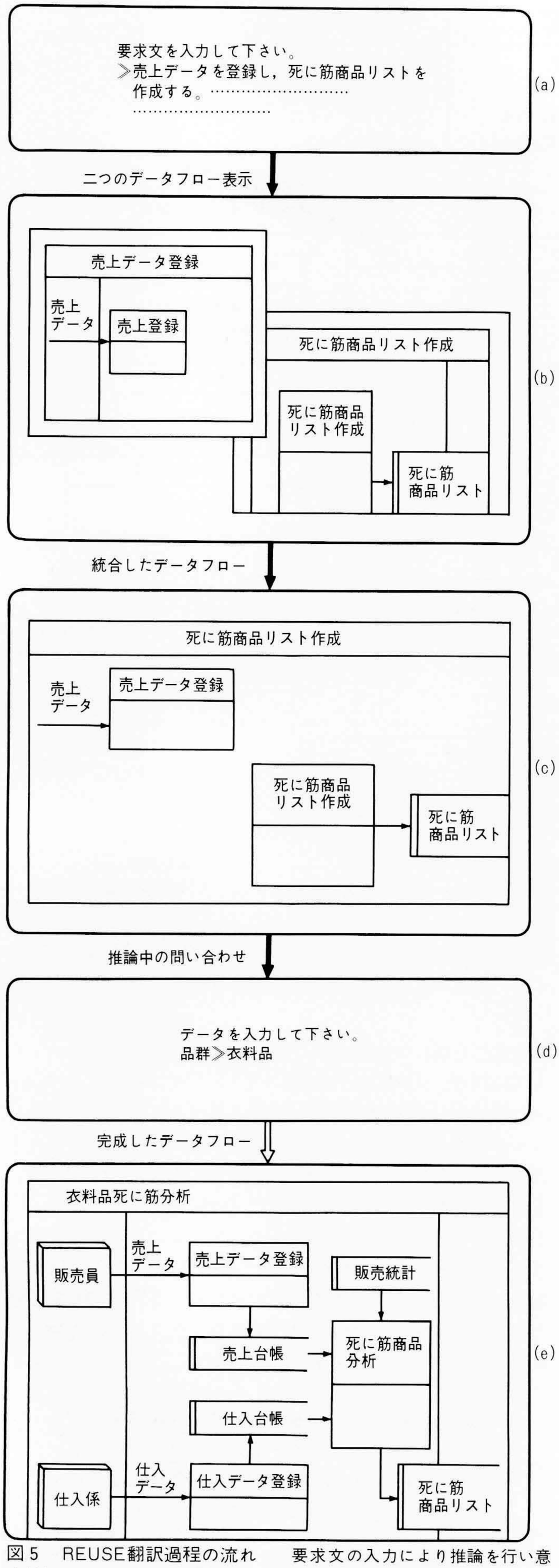

図5 REUSE翻訳過程の流れ 要求文の入力により推論を行い意味付加し，データフロー図をユーザーと対話しながら作成する。

5 適用効果

現在，REUSEはHITAC Mシリーズ VOS 3(Virtual-storage Operating System 3)上でプロトタイプを試用している。本開発で確立した基盤技術には次の特徴がある。

(1) システム要求仕様(承認仕様)設定の迅速化

粗い表現の要求から，類似システムの外部仕様を修正して顧客に提示できるようになり，従来の人手による方式と比べて，要求仕様設定のための工数，要求定義の漏れによる修正工数を大幅に改善できる。

(2) システム設計工数の低減

段階的な設計詳細化を，類似例を用いて実施できるため，従来方式に比べ端末操作数，端末作業時間，仕様記述と修正工数を大幅に改善できる。

(3) 仕様書再利用の促進によるソフトウェア資源の活用

システム開発のために作成された仕様書が，プログラムを含めて再利用の対象となる。既に開発したソフトウェアの仕様は，ソフトウェアを新たに開発する場合の資源とすることができる。既開発ソフトウェアのすべてがソフトウェア開発の財産となるならば，効果は図りしれない。

6 結 言

ICAS-REUSEは，仕様書の再利用によりシステムの設計を支援する。

要求を記述した文章を，日本語処理の技術を応用してシステム仕様に変換し，更に内蔵する業務の知識に基づいて不足情報を補い，既存仕様書を検索，再利用するものである。

ICAS-REUSEを用いることにより，ユーザーの言葉を用いて既存システム，ソフトウェアの仕様書を再利用することが可能となり，ユーザーの承認仕様設定の迅速化，システム設計者のシステム外部仕様設定の効率化，更に仕様書再利用によりソフトウェア設計の効率化などが実現できる。

今後は，要求文解読のための辞書，業務の概念依存関係などの知識の拡充，知識を用いる推論の高速化を図り，機能性及び操作性を更に向上させてゆく。

参考文献

1) 葉木，外：システム開発支援ソフトウェア“EAGLE”—EAGLE拡張版“EAGLE 2”，日立評論，**68**，5，373～378(昭61-5)
2) 葉木，外：システム開発支援ソフトウェア“EAGLE”，日立評論，**66**，3，189～194(昭59-3)
3) 小林，外：ソフトウェア一貫生産システム“ICAS”基盤技術の確立，日立評論，**68**，5，172～176(昭61-5)
4) 小林，外：プログラムの自動作成・再利用技術，電気・情報関連学会連合大会講演論文集，5-35～5-38(昭和60)
5) J.M.Neighbors：The Draco Approach to Constructing Software from Reusable Components, IEEE Tr.on Software Engineering, **SE-10**, 5, 564～574(Sept., 1984)
6) H.Kitagawa, et al.：Form Document Management System SPECDOQ-Its Architecture and Implementation, ACM-SIGOA, 132～142(June, 1984)