4L-3

ビジネスソフトウェアを対象とした設計支援エキスパートシステムの提案

田口　浩一，　千吉良　英毅　　岸　一也
((株)日立製作所　システム開発研究所)　(同　大森ソフトウェア工場)

1．はじめに

近年，ビジネス分野における事務処理（伝票発行，帳票作成等）の機械化が急速に進み，オフィスコンピュータの適用範囲が拡大し，顧客システムも複雑化かつ多様化している。計算機支援による開発者の負担軽減，生産性向上が，現在強く望まれ，特に上流工程を支援するための研究が盛んに行われている。

本稿では，ビジネス（事務処理）ソフトウェア開発におけるライフサイクルの中でいまだコンピュータ・スペシャリストの知識や経験に強く依存していると思われる業務処理設計，システム設計に中心を置き，これを支援するための設計支援エキスパートシステムについて報告する。

2．事務処理ソフトウェア設計支援に関する基本的考え方

設計者の仕様決定手順やノウハウに基づき設計活動を体系化し，これを知識データベースとして構築する。これに基づいた設計作業の知的支援機能，図形等を用いた仕様の表現方法及びその編集機能，仕様決定の際のガイダンス機能等，ユーザフレンドリーなインタフェースを備えた設計支援環境を構築し，設計作業の自動化を図り，生産性を高めていく。

3．事務処理ソフトウェアの仕様決定手順

3．1仕様変換の考え方

要求定義からコーディングに至るソフトウェア開発工程は，要求仕様や設計仕様等の各種仕様を経てソースコードに至る仕様変換過程と言える。これは，隣合う仕様間のセマンティックスギャップを埋めるための一連の手順である。事務処理ソフトウェアの仕様変換過程を，設計者の仕様決定手順やノウハウに基づいて図１に示すようにとらえた。要求仕様である業務フロー（ａ）をゼネラルフロー（ｂ）に変換し，これに基づいてシステムフロー（ｃ）を求め［１］，システムフローからプログラム仕様，更には，プログラムを生成する。設計支援エキスパートシステムは，この中で業務フローからシステムフロー生成に至る過程の支援を狙ったものである。このために，設計者の知識やノウハウを明確にした。上記の各仕様について解説する。

（１）業務フロー

ＤＦＤ［１］の機械化作業部分を定義し，対象業務のデータのやりとりを表現したフロー図である。

（２）ゼネラルフロー

複数ジョブ間のデータのやりとりを表現したフロー図である。

（３）システムフロー

ジョブステップ間のデータのやりとりを表現したフロー図である。

上記各フローにおいて，「業務」，「ジョブ」，「ジョブステップ」をＤＦＤのプロセス，及びプロセス内のデータのやりとりをデータフローと呼ぶこととする。また，あるプロセスがシーケンシャルに行う入出力において，データの読み込み，書き出し単位をそれぞれ「入力イベント」，「出力イベント」，入出力イベントの発生でやりとりされる情報の単位を「トークン」と呼ぶこととする。

3．2仕様変換知識の定式化

仕様変換過程を機械化し，自動化を図っていくためには，設計者の知識やノウハウに依存して行なわれてきた仕様変換手順における問題点を明確にする必要がある。その為に，業務フロー，ゼネラルフロー，システムフローをそれぞれ入力データ構造と出力データ構造の対応関係という観点［２］から整理した。又，各フローの変換過程における設計者の知識，ノウハウを分析した。その結果，次の点が明らかになった。

（１）「業務」，「ジョブ」，「ジョブステップ」において，入出力イベント同士の発生順序が非決定的であったり，出力データ作成のために，過去に入力されたデータを参照する必要がある。

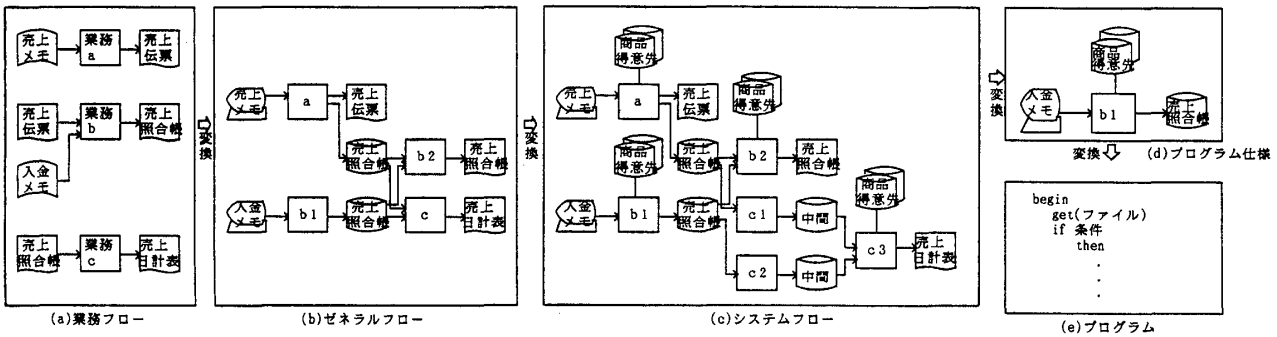

図１．事務処理ソフトウェアの仕様変換過程

"A Design Expert System for Business Aplication Software"
Kouichi Taguchi,Eiki Chigira,Kazuya Kishi
Systems Development Laboratory, Hitachi Ltd., Ohmori Software Works, Hitachi Ltd.

（2）設計者の知識やノウハウは，このような場合に，入出力イベントの時間的な順序関係やデータ変換を直接規定するものである。

本システムでは，この様な設計者の知識やノウハウを体系化し，仕様変換の知識として定式化した。

（1）業務フローの変換知識

複数の業務が存在し，各々が並行して進行するプロセスは，イベントの発生順序が非決定的となる。これを「脈絡の不一致」と呼ぶ。「脈絡の不一致」を持ったプロセスでは，業務の分割を行う必要がある。入出力イベントの対象プロセスPを，単一の「業務」，プロセスP_1，P_2…Pnに分割する。

（2）ゼネラルフローへの変換知識

出力イベントのトークンが入力イベントのトークンを参照している場合，お互いを参照関係にあると呼ぶ。参照関係にある入出力イベント間には，ファイルが必要となる。これを，「処理サイクルの不一致」と呼ぶ。「処理サイクルの不一致」を持ったプロセスでは，業務の分割を行う必要がある。入出力イベントの対象プロセスPを，入力イベントの発生するプロセスP_1と，出力イベントの発生するプロセスP_2に分割し，プロセスP_1とプロセスP_2の間にファイルを設ける。

（3）システムフローへの変換知識

一連の入力イベントにおけるトークン列と，一連の出力イベントにおけるトークン列の順序が時系列的に対応しない場合（入力トークンに対応する出力トークンが出力される前に次のトークンの入力があるような場合），入出力イベント間にファイルが必要となる。これを「順序の不一致」と呼ぶ。「順序の不一致」を持ったプロセスでは，入出力イベントの対象プロセスPを，入力イベントの発生するプロセスP_1と，出力イベントの発生するプロセスP_2に分割し，プロセスP_1とプロセスP_2の間にファイルを設ける。

（4）ファイル作成の知識

業務において参照関係にある入出力イベントの入力データ構造を持ったファイルを作成する。

4．設計支援エキスパートシステムの概要

4．1設計支援エキスパートシステムの構成

本システムは，図2に示すように四つの系から構成される。それぞれの系の概要を以下に示す。

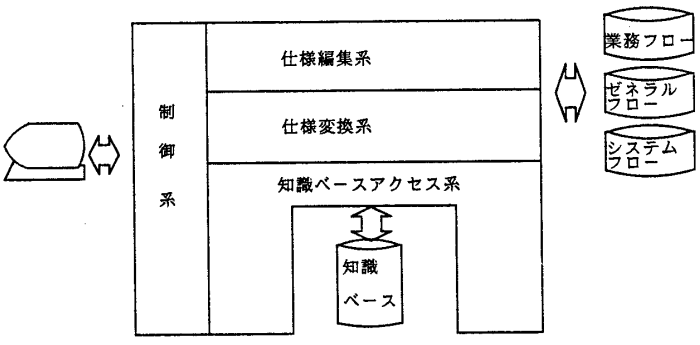

図2．設計支援エキスパートシステムの構成

（1）制御系

設計作業の作業メニュー，作業ガイダンスを備えたユーザとの対話機能を提供する。

（2）仕様編集系

仕様編集のための編集機能を提供する。対象業務分野の知識，フロー変換の為のルールを保持し，仕様の整合性を保証する。

（3）仕様変換系

仕様の変換を仕様変換ルールに従って推論実行するための推論系である。

（4）知識ベース管理系

知識ベースアクセス系，知識ベースから構成される。アクセス系は，知識ベースの作成，更新の為の機能を提供する。知識ベースは，設計仕様をフレーム形式で，対象業務分野の知識，仕様変換規則をルール形式で表現する。

4．2設計支援エキスパートシステムを用いた設計作業

設計作業は，制御系が提供する作業メニュー，作業ガイダンスに従い進める。まず，仕様編集系を通して対象業務の要求仕様を対話的に入力する。対象業務分野の知識を使って業務仕様を体系的に整理，検証して業務仕様の整合性を保証する。次に，仕様変換系を使って，ゼネラルフロー，システムフローを自動生成する。ユーザは，自分の意図と解釈された結果が一致しているかどうか仕様編集系を使って確認する。結果に不適当な部分がある場合には，通常，業務仕様の修正を行い再入力する。また，仕様編集系を使って直接修正することも可能である。ルールの矛盾は，知識データベースアクセス系を使って更新する。

5．おわりに

設計者の経験やノウハウに基づいた仕様変換方式を核とする設計支援エキスパートシステムについて報告した。本システムを使った設計作業の実現によって，以下のような効果が期待できる。

（1）対象業務の要求仕様記述が簡略化でき，要求項目の漏れを防止する。

（2）図形による仕様の表現，作業メニュー，作業ガイダンス，設計仕様の自動生成等により設計活動を支援する。

（3）業務分野知識，仕様変換規則により仕様の矛盾を早期に発見できる。

（4）設計者の個人差によらない設計作業の一定水準の品質が保証される。

【参考文献】

[1] DeMarco,T.:"Structured Analysis and System Specification". Prentice-Hall/Yourdon Press(1978)

[2] Jackson,M.A:"Principles of Program Design". Academic Press(1979)