The University of Osaka
Institutional Knowledge Archive

Title	分散処理システムにおけるジョブ管理方式に関する研究
Author(s)	小林, 隆
Citation	大阪大学, 1999, 博士論文
Version Type	VoR
URL	https://doi.org/10.11501/3155449
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

分散処理システムにおける
ジョブ管理方式に関する研究

1998年12月

小林　隆

分散処理システムにおける
ジョブ管理方式に関する研究

1998 年 12 月

小林　隆

内容梗概

本論文は、筆者が 1982 年から現在までに、(株) 日立製作所　システム開発研究所、ビジネスシステム開発センタ、システム開発本部において従事した分散処理システムに関する研究開発業務の経験を背景に、1997 年から現在にかけて大阪大学大学院工学研究科博士後期課程在学中に行ってきた、分散処理システムにおけるジョブ管理方式に関する研究成果をまとめたものである。

マイクロコンピュータとネットワーク技術の発展は、単独のコンピュータではできなかったより大きな業務を協同で実行することを可能とした。ある目的と制約条件をもった仕事（ジョブ）をいくつかの要素作業（タスク）に分割し、それらを複数の行為者（アクター）に割当てて実行する。このようなシステムを広く分散処理システムとよぶ。アクターとしては、計算機システム、人間、自動機械、等、種々の実行主体を想定する。分散処理システムの第一の狙いは、単独のアクターでは実行困難なジョブを複数のアクターの協同作業により実現することであり、そのためにジョブ管理方式の最適化が重要な課題となる。最適化は、タスク間の順序関係を定めるプロセス設計と、個々のタスクの実行方法を定めるスケジューリングに分類できる。分散処理システムの第二の狙いは、環境変化に対してタスクとアクターの組替えにより柔軟にシステム構造を変更して対処することであり、そのためにジョブ管理方式の柔軟性が課題となる。柔軟性は、ジョブ特性のばらつきに対応する短期柔軟性と、実行環境の時間的変化に対応する長期柔軟性に分類できる。ジョブ管理方式の設計では、最適化と柔軟性のトレードオフを行うことが最大の目的となる。

本研究では、上記の目的を実現するために次の２点を課題とする。

(1)相互に影響する複数の特性をもつジョブのスケジューリング問題の解決。

(2)環境変化に強い柔軟なジョブ管理システムの実現。

まず、第一の課題のために、計算機入出力装置の一種であるカートリッジ磁気テープライブラリ装置のアクセッサの干渉回避制御を対象として、段階的なヒューリスティックス適用方法を提案する。ここでは、システム工学のモジュール方式を適用して、干渉回避問題をドライブ装置選択とアクセッサ割当という独立性の高い２つのサブ問題に分割し、各々をヒューリスティックスにより解決し、最後に全体の調整を行う。さらに、ヒューリスティックス方式の評価を迅速かつ低工数で行うために、各サブ問題を予め静的に解析してシステム状態と処理性能の関数としてモデル化し、これらの結果を確率ペトリネットに

より統合してマルコフモデルに帰着させる。

次に、第二の課題のために、ワークフローシステムの構築技法を対象として、プロセス安定構造の抽出に基づくシステム構築方法を提案する。ここでは、タスクの順序制約の本質的で変化し難い安定構造を明らかにし、個別性の高い変化部分と明確に分離して実装を行うことによりメンテナンス性を向上する。さらに、ワークフローシステムを効率よく構築するために、標準的なビジネスプロセス仕様をワークフロー業務テンプレートとして予めライブラリ化し、それを個別仕様に合わせてカスタマイズするアプローチをとる。

本論文は全６章から構成される。

第１章の序論では、分散処理システムのジョブ管理問題を定義しその技術課題が最適化と柔軟性のトレードオフであることを述べ、これら２つの分類軸に従って、計算機工学、生産工学、経営工学の各分野におけるジョブ管理方式の研究動向を明らかにする。そして、その結果に基づいて、本研究の目的と課題を設定する。

第２章では、計算機入出力装置の一種であるカートリッジ磁気テープライブラリ装置のアクセッサの干渉回避制御方式を提案する。本装置は、テープ約 6,500 巻を収納する自動倉庫機能と、２台のアクセッサ（ロボットハンド）がドライブ装置最大 32 台に対してテープをマウントする自動搬送機能を有する。この装置の応答性と処理性能を向上するためには、２台のアクセッサが互いに他の動作を邪魔する、いわゆる、「干渉」を回避することが不可欠である。この問題を解決するために、システム工学のモジュール方式により、ドライブ装置選択とアクセッサ割当を独立に実施できるように問題を分割し、各々にヒューリスティックスを適用するアプローチをとる。

第３章では、共有リソースの競合によるジョブの干渉を回避するためのヒューリスティックなスケジューリング方式を対象として、その評価方法を検討する。従来の待ち行列理論は適用範囲が先入れ先出し、後入れ先出し等の単純な論理に限定されており、シミュレーションはプログラムの開発工数が膨大となる。これらの問題を解決するために、スケジューリング論理をシステム状態と処理性能の関数としてモデル化し、その性能関数を確率ペトリネットモデルに組込むことにより単純で実用的な評価モデルを実現する。そして、第２章で論じた磁気テープライブラリ装置の干渉回避制御方式の評価を事例として、その効果を示す。

第４章では、ワークフローシステムのメンテナンス性を向上するためのシステム構築技法を検討する。ビジネス環境の変化に柔軟に追従するワークフローシステムを構築するためには、システム仕様の安定した部分と変化しやすい部分を分離することが必要となる。この問題を解決するために、業務の委託階層に着目して、ビジネスプロセスの本質的な安

定構造である設計パターンを抽出し、それに基づいてシステムの設計と実装を行う。

第５章では、ワークフローシステム構築の生産性を向上するための、ワークフロー業務テンプレートについて検討する。第４章で議論した設計パターンに基づいて、種々のビジネスプロセスの標準仕様を抽出してライブラリ化し、それを個別仕様に合わせてカスタマイズすることによりワークフローシステムを設計する。さらに、設計結果を実装するための支援ツールを提案する。

第６章では、本研究で得られた成果を要約し、今後の研究課題を述べる。

研究業績

Ⅰ. 学会論文誌

1. 小林隆, 栗原謙三: プラント診断エキスパートシステムにおける段階的に異常原因を絞り込む推論方式の提案, 情報処理学会論文誌, Vol. 29, No. 11, pp. 1019/1027 (1988.11)
2. 栗原謙三, 小林隆, 明石吉三: 半導体プロセス診断エキスパートシステム, 情報処理学会論文誌, Vol. 30, No.7, pp. 918/921 (1989.7)
3. 栗原謙三, 原敬市, 小林隆, 汐見龍徳: 線型計画法の支援によるルールベース型作業スケジュール方式, 情報処理学会論文誌, Vol. 30, No. 8, pp. 976/989 (1989.8)
4. Kobayashi, T. and Kurihara, K.: Performance Analysis of Job Scheduling Policy for Interference Avoidance Using Stochastic Petri Nets, IEICE TRANSACTIONS, Vol. E 74, No. 10, pp. 3144/3151 (1991.10)
5. 小林隆, 栗原謙三, 里山愛, 小菅稔: カートリッジ磁気テープライブラリの制御方式, 情報処理学会論文誌, Vol. 33, No. 5, pp. 611/619 (1992.5)
6. 小野田仙一, 小林隆, 尾越昌子, 湯本真樹, 薦田憲久: 資材注文依頼業務のワークフローについてのターンアラウンドタイム分析, 電気学会論文誌, Vol. 118-C, No. 4, pp. 585/591 (1998.4)
7. 小林隆, 薦田憲久: 高メンテナンス性を指向したワークフローシステム設計技法, 電気学会論文誌、Vol. 118-C, No. 7, pp. 1130/1137 (1998.7)

Ⅱ. 国際学会

1. Kobayashi, T. and Kurihara, K.: Performance Model of Interference Avoidance Policy Using Stochastic Petri Nets, in Proc. of 1991 IEEE International Workshop of Petri Nets and Performance Models, Melbourne, Australia, pp. 125/133 (1991.12)
2. Kobayashi, T., Ogoshi, S. and Komoda, N.: A Business Process Design Method for Applying Workflow Tools, in Proc. of 1997 IEEE International Conference on System, Man, and Cybernetics, Orlando, Florida, USA, pp. 2314/2319 (1997.10)
3. Onoda, S., Kobayashi, T. and Komoda, N.: Bottleneck Detection Analysis for Workflow Improvement, in Proc. of 1997 IEEE International Conference on System, Man, and

Cybernetics, Orlando, Florida, USA, pp. 3331/3336 (1997.10)

4. Kobayashi, T., Ohmura, Y., Ogoshi, S., Sakai, K. and Komoda, N.: Development of Workflow Template for Administration Process, in Proc. of 1998 IEEE International Conference and Workshop on Engineering of Computer Based Systems, Jerusalem, Israel, pp. 290/297 (1998.4)
5. Maruta, T., Onoda, S., Ikkai, Y., Kobayashi, T. and Komoda, N.: A Deadlock Detection Algorithm for Business Processes Workflow Models, in Proc. of 1998 IEEE International Conference on System, Man, and Cybernetics, San Diego, California, USA, pp. 611/616 (1998.10)
6. Onoda, S., Ikkai, Y., Kobayashi, T. and Komoda, N.: Definition of Deadlock Patterns for Business Processes Workflow Models, in Proc. of the Thirty-Second Hawaii International Conference on System Sciences (HICSS-32), Wailea, Hawaii, USA (1999.1) (to appear)

III. 解説記事

1. 藤原久利, 阿部信雄, 阿部道雄, 小林隆, 高巣周平: 精密機械加工・組立工場におけるファクトリーオートメーション, 日立評論, Vol. 68, No. 11, pp. 31/36 (1986.11)
2. 西山淳, 関根章司, 黒須康雄, 小林隆: 書類のイメージ処理を可能としたワークフローシステム "TEAMSTAR", 日立評論, Vol. 77, No. 8, pp. 571/574 (1995.8)

IV. 国内研究会

1. 小林隆, 明石吉三, 小西洋三, 阿部道雄: 高精度機械製品生産における嵌合制御方式の提案, 昭和 60 年電気学会全国大会, pp. 2005/2006 (1985)
2. 小林隆, 栗原謙三, 明石吉三: 知識工学を応用した異常診断システムの高速推論方式の提案, 情報処理学会第 31 回（昭和 60 年後期）全国大会, pp. 951/952 (1985.9)
3. 小林隆, 栗原謙三, 明石吉三: 知識ベースに基づく半導体プロセス診断システムの機能構成, 情報処理学会第 33 回（昭和 61 年後期）全国大会, pp. 1181/1182 (1986.9)
4. 栗原謙三, 小林隆, 明石吉三: 知識ベースに基づく半導体プロセス診断システムにおけるマンマシンインタフェース, 情報処理学会第 33 回（昭和 61 年後期）全国大会, pp. 1295/1296 (1986.9)
5. 小林隆, 栗原謙三, 明石吉三: プラント診断エキスパートシステムにおける段階的に異常原因を絞り込む推論方式の提案, 情報処理学会第 35 回（昭和 62 年後期）全国大会, pp. 1577/1578 (1987.9)
6. 小林隆, 栗原謙三, 里山愛: カートリッジ磁気テープライブラリ装置制御方式の提案,

情報処理学会第 41 回（平成 2 年後期）全国大会, 第 4 分冊, pp. 79/80 (1990.9)

7. 小林隆, 栗原謙三: ジョブの干渉回避スケジューリング方策の確率ペトリネットを適用した性能解析, 計測自動制御学会 第 6 回離散事象システム研究会 講演論文集, pp. 69/76 (1990.12)
8. 里山愛, 小林隆, 栗原謙三: カートリッジ MT ライブラリ装置制御方式のシミュレーション評価, 情報処理学会第 42 回（平成 3 年前期）全国大会, 第 4 分冊, pp. 55/56 (1991.3)
9. 小林隆, 栗原謙三: ジョブの干渉回避スケジューリング方策の確率ペトリネットを適用した性能解析, 電気学会研究会資料（システム・制御研究会）, SC-91-8-13, pp. 1/9 (1991.8)
10. 尾越昌子, 小林隆, 大村義秀: ビジネスプロセスモデルを用いた業務分析設計技法の開発, 情報処理学会第 54 回（平成 9 年前期）全国大会, 第 1 分冊, pp. 383/384 (1997.3)
11. 大村義秀, 小林隆, 趙延悦: ビジネスプロセス設計パターンを用いたワークフローシステム設計開発技法の開発, 情報処理学会第 55 回（平成 9 年後期）全国大会, 第 4 分冊, pp. 173/174 (1997.9)
12. 小野田仙一, 小林隆, 薦田憲久: 資材注文依頼業務に対するワークフロー分析, 平成 9 年 電気学会電気・情報・システム部門大会, B-4-3, pp. 527/530（1997.9）
13. 小林隆, 小野田仙一, 薦田憲久: ワークフロー業務テンプレートの開発, 人工知能学会研究会資料（知識ベースシステム研究会）, SIG-KBS-9801, pp. 31/36 (1998.9)
14. 小野田仙一, 一階良知, 小林隆, 薦田憲久: ワークフローモデルにおけるデッドロックパターンの定義, 平成 10 年 電気学会電気・情報・システム部門大会, B-2-3, pp. 399/402（1998.9）
15. 圓田哲也, 小野田仙一, 一階良知, 小林隆, 薦田憲久: ワークフローモデルにおけるデッドロックパターン検出アルゴリズム, 平成 10 年 電気学会電気・情報・システム部門大会, B-2-4, pp. 403/406（1998.9）

目　次

第1章　序　論 1

1.1　研究の背景 1

1.2　従来研究 5

1.3　研究課題と方針 10

1.4　論文の構成 13

第2章　カートリッジ磁気テープライブラリの制御方式 15

2.1　緒言 15

2.2　ライブラリ装置のシステム構成 16

2.3　干渉回避問題 17

2.4　干渉回避制御方式 20

2.4.1　基本的考え方 20

2.4.2　制御方式 22

2.5　評価 26

2.5.1　評価条件 26

2.5.2　評価結果 28

2.6　結言 29

第3章　ジョブの干渉回避スケジューリング方式の確率ペトリネットを適用した性能解析 31

3.1　緒言 31

3.2　スケジューリング方式性能解析問題 33

3.3　確率ペトリネット適用解析方式 35

3.3.1　考え方 35

3.3.2　解析方式 35

3.4　磁気テープライブラリへの適用 38

3.4.1　制御方式の解析 38

3.4.2　提案解析方式の効果 42

3.5 結言 45

第4章 高メンテナンス性を指向したワークフローシステム構築技法 47

4.1 緒言 47

4.2 ワークフローシステム構築の課題 48

4.2.1 ワークフローシステム構築手順 48

4.2.2 ビジネスプロセス設計問題 52

4.2.3 メンテナンス性向上の課題 53

4.2.4 タスク階層モデル 54

4.3 メンテナンス指向ワークフローシステム構築技法 56

4.3.1 基本方針 56

4.3.2 プロセスパターンの抽出方法 57

4.3.3 情報パターンの抽出方法 59

4.3.4 設計パターンに基づくワークフローシステム構築手順 61

4.4 ケーススタディ 65

4.5 結言 70

第5章 間接部門の申請業務を対象としたワークフロー業務テンプレートの開発 71

5.1 緒言 71

5.2 ワークフロー業務テンプレートの要件 72

5.2.1 従来方式と問題点 72

5.2.2 ワークフロー業務テンプレートの要件 74

5.3 ワークフロー業務テンプレート 76

5.3.1 ビジネスプロセス標準仕様抽出の考え方 76

5.3.2 間接部門申請業務の設計パターン 76

5.3.3 ワークフロー業務テンプレートと実装支援ツールの全体構成 80

5.3.4 ワークフロー業務テンプレート 81

5.3.5 実装支援ツール 83

5.4 ワークフローシステム構築事例 86

5.5 効果 88

5.6 結言 90

第6章 結 論 93

謝　辞 97

参考文献 99

第1章　序　論

1.1　研究の背景

コンピュータは今やどこにでもある。シリコンチップは、家庭、学校、職場、レジャー等、いたるところで見出すことができる[1]。ネットワーク技術の発展はそれらのマイクロコンピュータを有機的に接続し、その結果、個々のコンピュータではできなかったより大きな業務を協同で実行することを可能とした。

ある目的と制約条件をもった仕事（ジョブ）をいくつかの要素作業（タスク）に分割し、それらを複数の行為者（アクター）に割り当て、予め定めた順序（オーダー）に従って実行する。このようなシステムを広く分散処理システムとよぶことにする。アクターとしては、計算機システム、人間、自動機械、等、種々の実行主体を想定する。分散処理システムの狙いは、単独のアクターでは実行困難なジョブを複数のアクターの協同作業により実現することである。そのために、「ジョブの目的を達成するために複数の分散タスクをどのように調整するか」というジョブ管理が重要な問題となる。図 1.1 に分散処理システムにおけるジョブ管理方式の位置付けを示す。

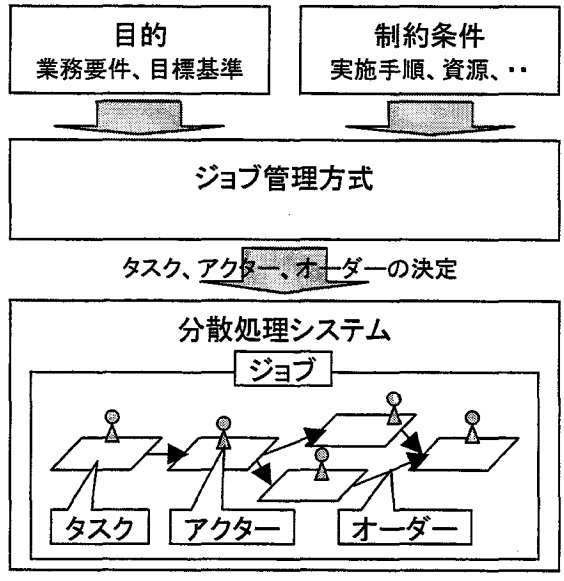


図 1.1　分散処理システムにおけるジョブ管理方式の位置付け

分散処理システムがいかに世の中に浸透し重要な役割を担っているかを理解するために、以下に、計算機工学、生産工学、経営工学の各分野における事例を示す。

計算機工学分野では、マイクロプロセッサ技術とネットワーク技術の発展に伴い、分散処理システムの研究と実用化が急速に進んでいる。分散処理システムは、通常、並列処理単位の粒度により下記の３種類に分類できる[2]。

①コンピュータネットワーク：パーソナルコンピュータ、メインフレーム等のスタンドアロン型のコンピュータが、コンピュータネットワークを介してつながったシステム。データベース、入出力機器等の管理をこれらのコンピュータが協力して行う。

②マルチプロセッサシステム：複数の汎用プロセッサが、共通の目的のために単一の制御機構により動作するシステム。プロセッサの接続関係と通信方法がシステムの特性を決定する重要な要素となる。

③専用計算機：特殊な目的のために開発された分散処理システム。スタンドアロン計算機の形態をとるが、入出力装置のように汎用計算機に接続して使用されることもある。行列演算、フーリエ変換等の特殊アルゴリズム処理ユニット、アレイプロセッサ、非ノイマン型計算機、等がある。

ジョブ管理という観点からみると、①のコンピュータネットワークでは、個々のプロセッサは独立に処理を行い、入出力装置、データベース等の限定された共有資源に関してのみ割当調整を行う。これに対して、②のマイクロプロセッサシステムでは、ある一つの目的に向けてシステム内の全ての資源を管理するため、プロセッサ間でより詳細な情報を交信しタスク群の実行状況に応じたスケジューリングを行う。③の専用計算機では、個々のプロセッサは極めて原始的なオペレーションのみを行い、他のプロセッサとの相互作用も限定されるため、ジョブ管理は単純で固定的なものとなる。

生産工学における分散処理システム技術は、1970 年代の FMS（Flexible Manufacturing System）の出現により発展した。FMS とは、柔軟性のある汎用設備群を計算機で制御し、多品種少量生産に対処する生産システムである。分散処理システムの構成要素は、マシニングセンター、自動搬送装置、自動倉庫、ロボット等のマイクロプロセッサを備えた自動設備とそれらを補完する作業者である。ここでのジョブ管理の役割は、部品、工程、生産量の多様化に対して、設備の変更を最小限に抑えて制御ソフトウェアの変更により対処すること、および、生産性を高めて決められた原価、生産量、納期等を遵守することである[3,4,5]。そのために、製品（ジョブ）を完成させるための工程計画、および、各工程における機械、作業者の日程計画が実施される。

経営工学における分散処理システム技術は、1980 年後半のグループウェア[6,7]の出現と1990 年前半の BPR（Business Process Reengineering）[8,9,10]の提唱により活性化した。特に、ワークフローは、ビジネスプロセスを自動化、管理するためのソフトウェアツールとして注目されている[11,12,13,14,15]。ワークフローは 1985 年に米国にて初めて製品化された。それまで、オフィスにおけるアプリケーションシステムは、ジョブを達成するために必要な個々のタスク機能の自動化を行っていた。その結果、ワードプロセッサ、帳票作成、スケジュール管理等の個人向けソフトウェアや、受注管理、会計管理、営業管理等の事務管理アプリケーションが数多くオフィスに導入された。しかし、タスク単位の自動化だけでは限定された効果しか得られず、オフィスの生産性向上は頭打ちになっていた。これに対して、ワークフローは、ジョブを構成するタスク群を電子的に統合すること、すなわち、業務手順の自動化を目的とする。そのために、分散しているアプリケーション、人間、自動機等をネットワーク技術によりつなぎ、ジョブ管理と情報伝達を行う。本ツールにより、人間系と計算機系を含む業務システムを分散処理システムのジョブ管理という観点から工学的に扱うことが可能となった。そして、ワークフローを適用して最適なジョブ管理を行うための各種のモデル化、分析、設計技術が開発されている。ここで議論されているテーマは、人間が勘と経験により実施してきた業務手順を分析して標準化すること、現行業務手順の問題点を解決し理想的なビジネスプロセスを設計すること、環境変化に応じてプロセス仕様を柔軟に変更すること、等である[16,17]。

表 1.1　分散処理システムにおけるジョブ管理問題

#	ジョブ管理問題の構成要素		内容	ジョブ管理方式の項目
1	目的		業務要件（成果物、資源）、 目標基準（スピード、品質、コスト）	
2	操作変数	タスク	与えられたジョブを複数人で分担して実行するための分割単位。	タスク分割
3		アクター	タスクの実行主体。計算機システム、人間、自動機械、等を想定。	スケジューリング
4		オーダー	タスク相互の順序関係。逐次、並列、同期、等がある。	プロセス設計
5	制約条件		調達可能なアクターと資源、目標基準の下限、実施手順、等に関する制約。	

分散処理システムにおけるジョブ管理問題を以下に定義する（表 1.1 参照）。複数の行為者が協同で実行すべき業務要件（期待する成果、投入する資源等）とその目標基準（ス

ピード、品質、コスト等）が、対象システムの外部から予め与えられるものとする。これをジョブと呼ぶ。与えられたジョブを実行するために操作可能な変数は、タスク、アクター、オーダーである。タスクはジョブを分担して実行するための分割単位であり、アクターはタスクの実行主体、オーダーはタスク相互の順序関係である。ジョブを実行する上では、資源、目標基準、実施手順、等に関する制約を遵守する必要がある。

このように、ジョブ管理問題は、「与えられた目的を制約条件の範囲内で達成するためにタスク、アクター、オーダーの３つの変数を決定する最適化の問題」と捉えることができる。ジョブ管理方式は、この最適化問題を解決するための技法であり、次の３つの項目からなる。

①タスク分割：与えられたジョブを、各アクターが実行する単位に分割する。この際、ジョブの目標基準を満足するために各タスクが遵守すべきサブ目標を設定する。目標基準として複数の項目が与えられた場合、各項目の優先順位、重み付け等を調整する必要がある。計算機工学にけるソフトウェアのモジュール化設計[18]、生産工学における WBS（Work Breakdown Structure）設計[19]、等が該当する。

②プロセス設計：ジョブを構成するタスク群の順序関係を決定する。計算機工学における分散アルゴリズムの設計[20]、生産工学における工程計画[3]、経営工学におけるプロジェクト計画[19]、ワークフロー定義[11]、等が該当する。

③スケジューリング：各タスクに対してアクターを割り当てる。また、各アクターが実行すべきタスクを複数個所有する場合の優先規則（ディスパッチング規則）を決定する。さらに、タスクを実行するために必要な共有資源を分配する。計算機工学におけるプロセッサスケジューリング[2]、生産工学における日程計画、機械スケジューリング[3]、経営工学におけるワークフローの作業者割当[11]、等が該当する。

ジョブ管理方式に対する要件としては、先に述べた最適化とともに、環境変化が生じても与えられた目的を達成できる柔軟性が重要である。これにより、ジョブ特性のばらつき、実行環境の時間的変化に対して、タスクとアクターの組替えにより柔軟にシステム構造を変更して対処する。柔軟性は、その時間スケールにより次の２つに分類できる[4]。

① 短期柔軟性：分散処理システムが与えられた時間内に扱うことができるジョブ特性のばらつき範囲を示す。ある特殊なジョブだけを扱うものは短期柔軟性が低く、種々の特性を持つ広範囲のジョブを扱うものは短期柔軟性が高い。マルチプロセッサスシステムのように、対象とするジョブの処理時間、必要資源、到着時間、等を一意に決めることが困難なものは、ジョブ管理方式の短期柔軟性が高くなければならな

い。

② 長期柔軟性：分散処理システムが実行環境（ジョブ特性、制約条件等）の時間的変化に追従できる範囲を示す。環境変化に伴うジョブ管理方式の変更が困難なものは長期柔軟性が低く、少ない工数で容易に変更可能なものは長期柔軟性が高い。FMSのように、マーケットニーズの変化に伴う製品仕様の変更に柔軟に対処する必要のあるものは、ジョブ管理方式の長期柔軟性が高くなければならない。

以上に述べたように、分散処理システムのジョブ管理方式の課題は、次に示す最適化と柔軟性である。

①最適化：定められた制約条件下での目的の達成

与えられたジョブ管理問題の解を見出すこと。すなわち、タスク分割、スケジューリング、プロセス設計を実施すること。

②柔軟性：実行環境の変化への柔軟な対応

問題環境の変化に柔軟に追従し常に最適解を提供すること。すなわち、短期柔軟性と長期柔軟性を実現すること。

最適化と柔軟性はともに重要な課題であるが、しばしば競合するものとなる。一般に、変化に柔軟に対応できる汎用的な方式は、ある条件に限定した専用方式に比べて最適な解を得ることが困難である。ジョブ管理方式の設計では、将来の変化を適切に予測しその範囲で最適化を検討すること、すなわち、最適化と柔軟性のトレードオフを行うことが最大の目的となる。

1.2 従来研究

上記の最適化と柔軟性の２つの課題を軸として、分散処理システムのジョブ管理方式の研究動向を分析する。最適化の軸は、ジョブ管理方式の構成要素であるプロセス設計、スケジューリングのどちらに関心があるかを示す。なお、タスク分割はプロセス設計に含めて考える。また、柔軟性の軸は、長期柔軟性と短期柔軟性のどちらに関心があるかを示す。このような２つの分類軸に従って、ジョブ管理方式の研究動向を下記のような４象現に分けて考える（図 1.2 参照）。

(1) 局所的汎用システム（最適化：スケジューリング、柔軟性：短期柔軟性）

対象とするジョブを構成する個々のタスクの実行方法にのみ関心があり、ジョブ特性の多様性に対処する必要があるシステム。マルチプロセッサシステムのように、ジョブ

の実行手順は単純であるが、処理時間、必要資源、到着頻度等のばらつきが大きいシステムが該当する。

(2) 局所的専用システム（最適化：スケジューリング、柔軟性：長期柔軟性）

(1)と同様に個々のタスクスケジューリングに関心がある。しかし、ジョブ特性のばらつきは小さく、むしろ、時間的な変化に対処する必要があるシステム。FMS におけるジョブショップのように、特定製品の製造工程の一部を担当し、製品仕様の時間的移り変りに追従するシステムが該当する。

(3) 全体的専用システム（最適化：プロセス設計、柔軟性：長期柔軟性）

局所的なタスクスケジューリングよりは、プロセス全体の実行計画に関心がある。そして、(2)と同様に時間的変化に柔軟に対処する必要があるシステム。ワークフローシステムのように、個々のタスクの実行方法は人間に一任し、その実行指示だけを行うシステムが該当する。ここでは、ビジネス環境の変化に対応して、柔軟にプロセスの変更を行えることが必要となる。

(4) 全体的汎用システム（最適化：プロセス設計、柔軟性：短期柔軟性）

(3)と同様にプロセス全体の実行計画に関心がある。しかも、ジョブ特性の時間的変化よりは多様性に対処する必要があるシステム。このような課題をもったシステムは現在見当たらない。将来的に、ワークフローシステムによりオフィスの合理化が進み、単純作業の完全自動化が実現された場合には、このようなシステムが必要となる可能性がある。

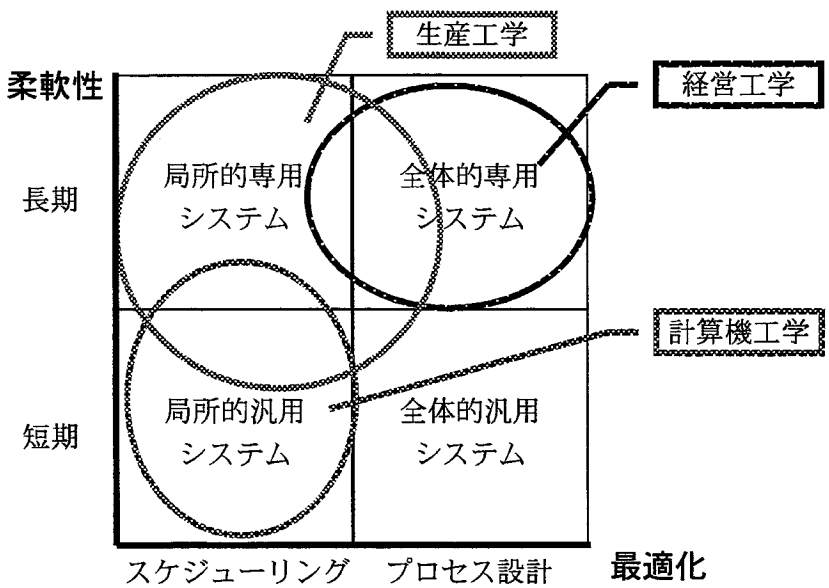


図 1.2 最適化と柔軟性によるジョブ管理方式の研究動向の分析

以下、図 1.2 に従って、計算機工学、生産工学、経営工学の各技術分野におけるジョブ管理方式の技術動向を説明する。

(1) 計算機工学分野

計算機工学で対象とするオペレーティングシステム、入出力制御システム等では、ジョブの実行手順は単純であるためプロセス設計は特に問題とならない。しかし、マルチ構成のプロセッサ、メモリ、データ転送パス等のリソーススケジューリングは、要求された性能を達成するために重要な課題となる。

一方、計算機システムが分野に依存しない汎用的なツールであり、ジョブの特性にかかわらず常に適切に動作することが義務づけられているため、ジョブ特性の多様性は大きい。これに対して、この分野における製品のライフサイクルは短命であるため、長期柔軟性は比較的小さくてよい。例えば、ディスク制御システムでは、データベースのバックアップ処理のように連続した記憶領域を長時間アクセスするものから、オンライン処理のように記憶領域をランダムにアクセスするものまでを混在して処理する必要がある。一方、システムのアーキテクチャは、プロセッサ、メモリ、チャネル等の性能向上に伴い１年～２年周期で変化する[21]。ジョブ管理方式が時間変化の追従性をもつ必要はなく、アーキテクチャが変わる毎に新たなジョブ管理方式を開発すればよい。以上をまとめると、計算機工学分野におけるジョブ管理方式の課題は、図 1.2 における局所的汎用システムに該当する。

このような背景から、計算機工学分野のジョブ管理方式の研究は、プロセッサ、メモリ等のスケジューリングに関するものが中心であり、ジョブ特性（処理時間、メモリ使用量、メモリアクセス頻度等）のばらつきをどのようにモデル化するかが課題となる。一般に、汎用的な目的に使用することを想定して、ジョブ特性のあらゆる可能性について考慮して最適化することはほとんど不可能である。この問題を解決するために、通常、確率理論を適用した近似的アプローチをとる[22,23]。従来、大型計算機システムの研究開発分野では、待ち行列ネットワークを中心とした確率理論を適用して、実用的なジョブ管理方式の開発が行われていた[24]。しかし、マルチプロセッサ構成の分散処理システムでは、システム構造、ジョブ特性、ジョブ間の相互作用、等が複雑に影響し合い、システムの全体最適化を解析的に行うことは難しくなった。従って、ジョブ管理方式の開発は、主としてヒューリスティックス（heuristics/発見法）に頼るようになった。

ヒューリスティックアプローチは、基本的には、人間の経験と勘および試行錯誤に基づくものであるが、効率よく準最適解を見出すためにはシステマティックな方法論、モデル、評価ツールが必要となる。従来、このような観点から研究が行われているが、実用的なものは少ない。研究レベルでは、マルチプロセッサシステムのアーキテクチャとタスクスケ

ジューリング方式を、確率ペトリネットの適用により設計する事例がある[2]。また、実用レベルでは、磁気テープ制御装置のマルチプロセッサ構成の設計を、近似的な待ち行列アプローチを適用して行った例がある[25]。

(2)生産工学分野

生産工学分野では、ジョブの実行手順は工程計画フェーズで、スケジューリングは日程計画フェーズで検討する。ジョブの実行手順は、IE（Industrial Engineering）の分野で古くから研究され、PERT（Program Evaluation and Review Technique）等のネットワーク計画技法により最適工程設計を行う方法が確立されている[19]。これに対して、スケジューリングは、古くから研究が行われいるにもかかわらず理論的な解法が開発されていない。生産工学分野におけるスケジューリング問題は、多品種少量の製品と多種複数の生産設備を対象とし、ジョブの時間的変化、作業時間の変動、工具の摩耗と破損、等のパラメータを考慮し、しかも、納期、設備稼働率、費用、等の複数の目標を実現する、極めて複雑な組み合せ最適化問題である[3]。このような複雑な問題について最適解を求めることは不可能に近い。従って、生産工学分野の最適性に関する課題は、対象システムの特性に応じた個別のスケジューリング方式を迅速かつ確実に開発することである。

一方、短期柔軟性は計算機工学のそれに比べて小さくてよい。１つの生産ラインが対象とする製品群は、部品、設備、製造手順等が共通する狭い範囲に限定されるからである。これに対して、生産ラインが市場変化に追従することは不可欠であり、ジョブ管理方式の長期柔軟性が重要な課題となる。以上をまとめると、生産工学分野におけるジョブ管理方式の課題は、図 1.2 における局所的専用システムに該当する。

このような理由から、生産工学分野のジョブ管理方式の研究は、計算機工学分野と同様に、ヒューリスティックスとシミュレーション評価を組み合せたアプローチがとられる[4]。例えば、生産工学分野のジョブ管理で決定すべき重要な事項の１つであるディスパッチングを考える。これは、処理すべきジョブが設備の前でいくつか待ち状態になっているとき、それらを設備に投入する順序を決めることである。代表的なディスパッチング規則には先着順、最小作業時間順、最早納期順、等があるが、実用的には、対象システムの特性に合わせてこれらの規則を組み合せたヒューリスティックな規則を作成する。そして、その規則が適切か否かを評価するためにシミュレーション技術を用いる。

以上に述べた最適性の問題よりもさらに重要視されているのは、ディスパッチング規則のメンテナンス性である。生産現場では、一度規則が作成されると、必ずしも適切ではない状況に変わってもそれが適用され続ける。その結果、当初は生産目標を達成する生産ラ

インであったにもかかわらず、時間の経過とともに納期遅れ、在庫超過、設備効率低下、等を引き起こす。この問題を解決するために、ジョブ特性、設備レイアウト、ディスパッチング規則等を容易に変更することができるシミュレーション環境の研究開発が行われている。例えば、ジョブ特性、設備レイアウトのモデル化にペトリネットを活用し、ディスパッチング規則のモデル化にルールベース技術を適用することにより、容易にシミュレーションモデルを開発する研究が行われている[26,27,28]。

(3)経営工学分野

経営工学分野におけるワークフローは、プロセス設計と管理を行うためのツールである。この分野のジョブの実行主体は未だ人間であり、計算機システムの役割は作業者を支援するためのツールの範囲に留まっている。従って、このような状況でジョブの実行管理に求められるのは、人間を効率よく働かせるためのスケジューリング方式ではない。プロセスの進行状況を監視し、適宜作業者にその情報を提供するジョブ管理システムの構築である。

一方、柔軟性は短期柔軟性、長期柔軟性ともに大きい。しかし、短期柔軟性に関しては、今のところ作業者の運用によって充分に対処できている。従って、現状の課題は、ジョブ管理システムの長期柔軟性の確保である。以上をまとめると、経営工学分野におけるジョブ管理方式の課題は、図 1.2 における全体的専用システムに該当する。

以上の理由により、ワークフローシステムに求められる要件は次の２点である。

①与えられた目標基準を達成するためのプロセス設計と管理。

②ビジネス環境の変化に追従できる柔軟性の確保。

上記①については、コンサルティングファーム、大学を中心として研究開発が進められ、IDEF (Integrated Definition for Function Modeling)、ユースケース等の分析設計技法が数多く発表されている[29,30,31,32,33]。また、シミュレーション技術を適用してプロセスのボトルネックやデッドロックを分析する研究が行われている[34,35,36,37]。一方、上記②については、ワークフローベンダーを中心に研究開発が進められ、ビジネスプロセス定義、アクター割当方法、アプリケーションインタフェース仕様、等を柔軟に変更可能なワークフロー製品が発売されている[14,15]。しかし、ワークフローツールとアプリケーションを含むワークフローシステム全体の柔軟性を向上する技術の開発はまだ少ない。ERP (Enterprise Resource Planning) との連携、オブジェクト指向技術の適用、等がそのブレイクスルーとなる可能性を秘めている[38,39,40,41]。

経営工学分野における分散処理システムのジョブ管理方式と類似した研究に、MIT の Malone らが提唱したコーディネーション理論がある[42,43,44]。コーディネーション理論で

ベースになっている分野はCSCW（Computer Supported Cooperative Work）の技術分野であり、計算機科学、経済学、組織学等における種々の理論と比較することにより、人間の協同作業のモデル化と分析を行うことに関心がある。Malone のコーディネーション理論では、下記の４項目が基本要素とされている。

・目的（Goals）

・活動（Activities）

・行為者（Actors）

・相互作用（Interdependencies）

このコーディネーション理論の中心的な関心事は、協同作業を管理する基本手順（コーディネーションプロセス）を汎用的なモデルとし、それを新たな問題の解決に適用することである。コーディネーションプロセスとしては、ゴール分割、資源分配、タスク順序設定、等を想定している。本研究で対象とする分散処理システムのジョブ管理方式と比較すると、協同作業の管理方式を統一的に扱うという方向性は同じである。しかし、コーディネション理論で主として議論されているのはジョブ管理方式の最適アルゴリズムであり、本研究で焦点としている最適化と柔軟性のトレードオフに関する研究はない。

1.3 研究課題と方針

本研究では、分散処理システムにおけるジョブ管理方式の最適化と柔軟性の問題を解決するために、前節で述べた従来研究の動向をふまえて、次の２つの課題を設定する。そして、以下に示す方針によりそれらを検討する（図 1.3 を参照）。

(1) 局所的汎用システムを対象とした、相互に影響する複数の特性をもつジョブのスケジューリング問題の解決。

(2) 全体的専用システムを対象とした、環境変化に柔軟に追従するジョブ管理システムの実現。

まず、第一の課題のためには「段階的ヒューリスティックス適用とその評価」という方針をとる。1.2 節で述べたように、主に計算機工学分野で対象とする局所的汎用システムでは、ジョブ特性の変動が大きい状況でのスケジューリング方式の最適化が課題となっている。一般に、このような問題に対しては確率モデルが適用されるが、確率的スケジューリング問題の最適解を理論的に求めることは困難である。そこで、多くの場合、ヒューリスティックスを適用して準最適解を求めるアプローチがとられている。想定されるジョブ特性の範囲で、与えられた目標基準を達成するようなスケジューリング方式を開発するた

めには、システマティックな方法論、モデル、評価ツールが必要となる。

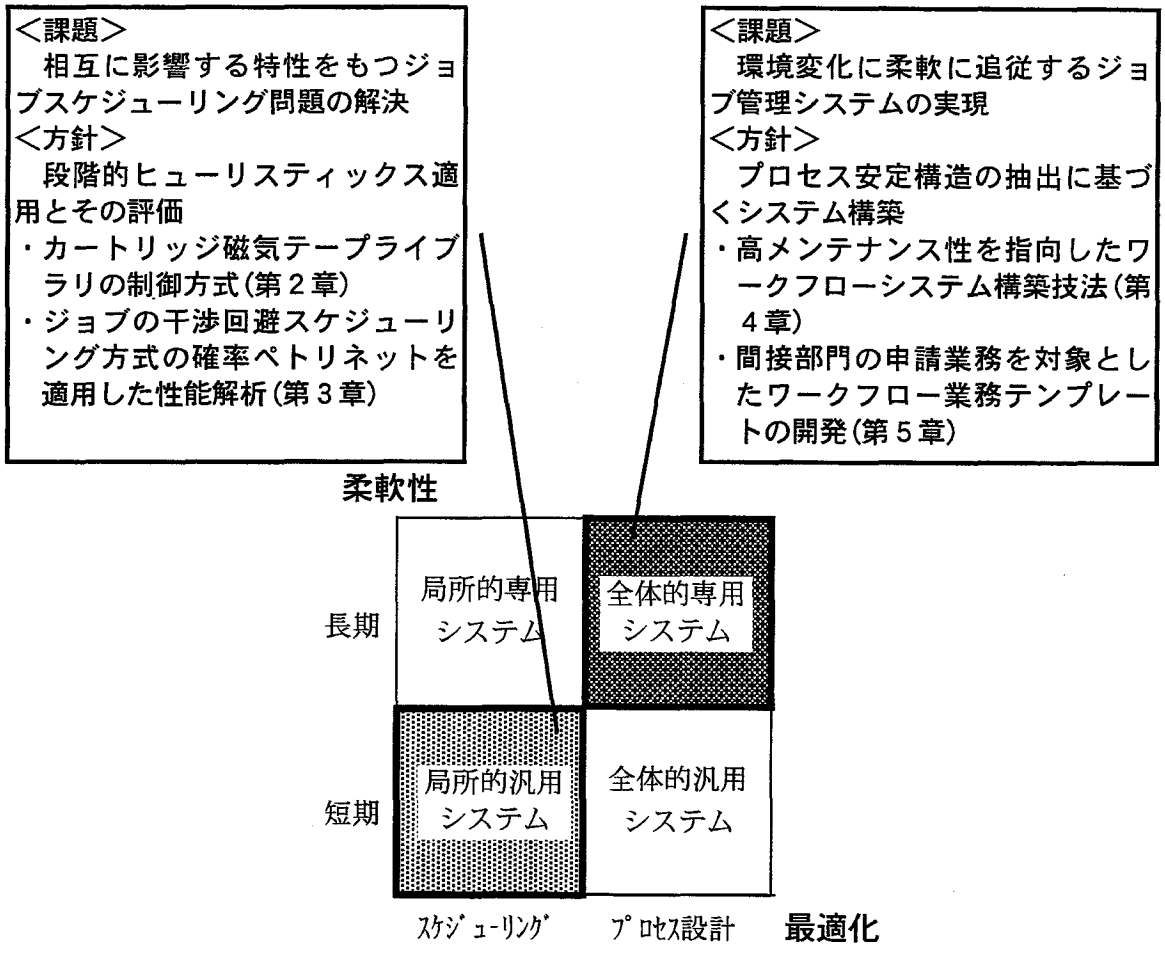


図 1.3　分散処理システムにおけるジョブ管理方式の研究課題と方針

本論文では、計算機入出力装置の一種であるカートリッジ磁気テープライブラリ装置のジョブスケジューリング方式を対象として、段階的なヒューリスティックス適用による方法を提案する。カートリッジ磁気テープライブラリ装置は、テープ約 6,500 巻を収納する自動倉庫機能と、2台のアクセッサ（ロボットハンド）がドライブ装置最大 32 台に対してテープをマウントする自動搬送機能を有する。本装置の応答性と処理性能を向上するためには、2台のアクセッサが互いに他の動作を邪魔する、いわゆる、「干渉」を回避することが不可欠である。本論文では、この干渉回避問題に対して、システム工学のモジュール方式を採用する。モジュール方式は、①対象システムのサブシステム分割、②各サブシステムの部分最適化、③全体としての調整、という3つの手順からなる[3]。この方式を適用して、干渉回避問題をドライブ装置選択とアクセッサ割当という独立性の高い2つの問

題に分割し、各々をヒューリスティックスにより解決し、最後に全体の調整を行う。このようなアプローチをとることにより、スケジューリング方式の論理を単純で直観的に理解できるようにし、致命的なミスを事前に防ぐことを目指す。

さらに、ヒューリスティックなジョブ管理方式を実際のシステムに適用する際には、その効果と限界を厳密に評価することが課題となる。既に述べたように、ヒューリスティックス方式の評価は、従来の待ち行列理論によるアプローチでは困難であり、シミュレーションに頼っているのが現状である。ところが、シミュレーションによる評価は適用範囲が広いが、プログラムの開発工数が膨大になるという問題がある。そこで、共有リソースの競合によるジョブの干渉を回避するためのスケジューリング方式を対象として、スケジューリング問題を構成するサブ問題を予め静的に解析し、システム状態と処理性能の関数としてモデル化し、確率ペトリネットを適用してそれらの結果を動的に統合するアプローチをとる。これにより、シミュレーションよりも少ない工数でヒューリスティックス方式の評価を行う。

次に、第二の課題のためには「プロセス安定構造の抽出に基づくシステム構築」という方針をとる。1.2 節で述べたように、主に、経営工学分野で対象とする全体的専用システムでは、ジョブ特性や環境条件が時間的に変化する状況下でのプロセス設計の最適化が課題となっている。このような課題を解決するためには、一般に、ジョブ管理方式の保守環境を充実する方法がとられる。ワークフローツールはその代表的な例であり、ジョブを構成するタスク群の実行順序やアクター割当のロジックをアプリケーションの処理仕様と分離して定義可能とし、従来のアプリケーションシステムに比べて高いメンテナンス性を実現している。しかし、個々のアプリケーションの処理仕様は、必ずしも、タスクの実行順序やアクター割当と独立に決められない。従って、ワークフローツールのメンテナンス性の高さは、ワークフローツールとアプリケーションを含むワークフローシステム全体のメンテナンス性を保証するものではない。

本論文では、タスクの順序制約の本質的で変化し難い安定構造を明らかにし、それに基づいてアプリケーションの処理仕様を決定するアプローチをとる。すなわち、プロセス安定構造に基づいて、ワークフローシステム仕様の安定した部分と変わりやすい部分を明確に分離して構築することにより、そのメンテナンス性を確保する。

さらに、メンテナンス性の高いワークフローシステムを効率よく構築するための生産性向上方式を議論する。ワークフローシステムの開発では、対象業務を分析してビジネスプロセス仕様を設計することに大きな工数を要する。そこで、プロセス安定構造に基づいて、標準的なビジネスプロセス仕様をワークフロー業務テンプレートとして予めライブラリ化

し、それを個別仕様に合わせてカスタマイズするアプローチをとる。そして、設計結果を実装するための支援ツールを提案する。

1.4 論文の構成

本論文は、第２章以降を次のように構成する。

第２章では、カートリッジ磁気テープライブラリ装置のアクセッサ群の干渉を回避するための制御方式を検討する。ここでは、システム工学のモジュール方式により、ドライブ装置選択とアクセッサ割当を独立に実施できるように問題を分割し、各々にヒューリスティックスを適用するアプローチをとる。この方法により、ヒューリスティックなスケジューリング方式の論理を単純で直観的に理解できるようにする。

第３章では、メインリソース群とサブリソース群により構成する典型的な分散処理システムを対象として、ジョブの干渉回避スケジューリング方式の性能解析を検討する。ここでは、従来の待ち行列理論とシミュレーションの問題点を解決するために、スケジューリング論理をシステム状態と処理性能の関数としてモデル化し、その性能関数を確率ペトリネットモデルに組込む。この方法により、実用的な時間と工数で解析可能な評価モデルを得ることができる。そして、第２章で論じる磁気テープライブラリ装置の干渉回避制御方式の評価を事例として、その効果を示す。

第４章では、ワークフローシステムのメンテナンス性を向上するためのシステム構築技法を検討する。ビジネス環境の変化に柔軟に追従するシステムを構築するためには、システム仕様の安定した部分と変化しやすい部分を分離することが必要となる。この問題を解決するために、業務の委託階層に着目して、ビジネスプロセスの本質的な安定構造である設計パターンを抽出し、それに基づいてシステムの設計と実装を行う。

第５章では、ワークフローシステム構築の生産性を向上するための、ワークフロー業務テンプレートについて検討する。第４章で議論した設計パターンに基づいて、種々のビジネスプロセスの標準仕様を抽出してライブラリ化し、それを個別仕様に合わせてカスタマイズすることによりワークフローシステムを設計する。さらに、設計結果を実装するための支援ツールを提案する。

第６章では、本研究で得られた成果を要約し、今後の研究課題を述べる。

第2章　カートリッジ磁気テープライブラリの制御方式

2.1　緒言

本章では、段階的なヒューリスティックス適用によるジョブ管理方式の開発方法を検討する。具体的には、計算機入出力装置の一種であるカートリッジ磁気テープライブラリ装置を題材として、共有リソースの競合回避を目的としたジョブ管理方式の開発方法を議論する[45,46,47]。

金融、証券、電力など各種の分野において、大規模なオンラインシステムの開発が進んでいる。それに伴い、データベース、そのバックアップデータ、あるいは、ジャーナルなどのデータ保管量が急速に増大している。多くの利用者は、比較的使用頻度の高い少量のデータ、例えば、データベースの最新データをアクセス速度の速い磁気ディスクに記憶し、バックアップデータを始めとするその他の大部分のデータを磁気テープに記憶している。磁気テープを使用するメリットは、記憶コストがその他の記憶媒体と比較して桁違いに安い点にある。デメリットは、テープの検索および掛け替え操作が人手により行われるために多大な労力を要する点にある。ところで、最近、計算機のデータ処理量が増大するのに伴い、このデメリットが急激にクローズアップされてきた。すなわち、1日あたりに利用するテープ量は数千巻にもおよび、テープ操作の要求はピーク時に数 10 秒に1回の割りで発生するようになってきた。テープ操作は、今や、作業員に対して精神的苦痛と肉体的疲労感を与える超過酷な労働と化している。

このような背景から、カートリッジ磁気テープ約 6500 巻を収納する自動倉庫機能と、ドライブ装置最大 32 台に対してテープを自動的にマウントする機能とを有するライブラリ装置を開発した。本ライブラリ装置は、オンラインシステムの中核となるべきものであるため、まず、無停止という重要な使命を担っている。さらに、応答性能と処理性能は少なくとも現状以上のレベルを確保しなければならない。このような高信頼性および高性能確保のための一手段として、本ライブラリ装置では、テープのハンドリングのための機構（以降、アクセッサとよぶ）を二重化し、しかも、それらを並列動作させる。ただし、設

置スペース、製造コストなどの制約により、２台のアクセッサがライブラリ内に設置された１本のレール上を走行する構造としている。

さて、２台のアクセッサが互いに他を邪魔することなく独立に動くことができれば、アクセッサ１台による運用時に比べて２倍の性能が発揮できる。ところが、実際は、前述したように１本のレール上を走行する構造であるため、各アクセッサは完全に独立には動けない。運搬経路の決め方、あるいは、運搬作業の割当方法により、一方のアクセッサが他の動作を邪魔するいわゆる「干渉」が多発してしまい、１台分程度の性能しか発揮できないことがある。並列動作による効果を向上するためには、このようなアクセッサ間の干渉を低減することが重要な課題となる。

本章では、この課題を解決するために、ホスト計算機から要求された磁気テープに対して、最適なドライブ装置とアクセッサを割り当てる干渉回避制御方式を提案する。また、シミュレーションにより定量的な効果を示す。

2.2 ライブラリ装置のシステム構成

本ライブラリ装置のシステム構成を、図 2.1 に従って説明する。

(1) 収納セル

カートリッジ型磁気テープを１巻ずつ収納するための棚。後述するアクセッサのレールをはさんで２列に並べて設置する。約 800 巻分の収納セルを１単位としてこれをフレームとよび、最大８フレームまで増設可能である。

(2) ドライブ装置

フレームの裏側に設置する。１フレームあたり最大４台のドライブ装置を設置可能である。収納セルの間に設けた小窓を介して、ドライブ装置にテープを受け渡すことができる。

(3) アクセッサ

ライブラリ内の指定された２点間でテープを運搬するためのハンドリング機構。ライブラリの長手方向に設置したレール上を垂直柱が走行し、その垂直柱に沿ってロボットハンドが走行する。設置スペースの制約のため、２台のアクセッサが１本のレール上を走行する構造となっている。このような構造のため、アクセッサが互いに干渉しあう可能性が高い。例えば、レールの左側に位置するアクセッサが右端の動作を行なっている最中には、右側に位置するアクセッサは動作を開始できない。

(4) ライブラリ・コントローラ

ホスト計算機のオペレーティング・システムから、運搬すべきテープ番号、使用するド

ライブ装置番号、および、マウント／デマウントの区別を受付け、各アクセッサに対して動作シーケンスを指示する。この際、２台のアクセッサの干渉を回避するための制御が必要となる。

(5) ＭＴ（Magnetic Tape）コントローラ

ライブラリ装置内のドライブ装置とホスト計算機との間のデータ転送を制御する。

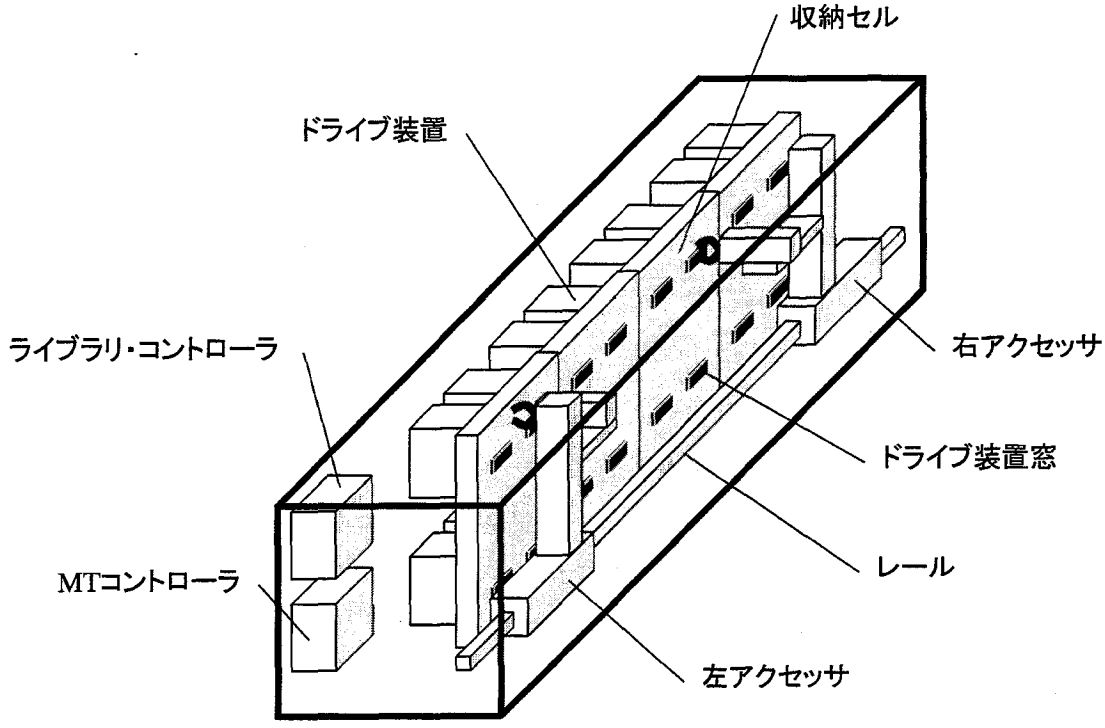


図 2.1　ライブラリ装置のシステム構成

2.3　干渉回避問題

まず、アクセッサが実行すべきジョブを定義する。テープを把持する位置を始点とよび、解放する位置を終点とよぶことにする。アクセッサの運搬ジョブとは、待機点から始点までの空走行と始点から終点までの実走行を合わせたものとする。次に、アクセッサの干渉とは何かを定義する。２台のアクセッサが安全上予め定められた距離限界より接近する可能性がある場合、これを回避するために、いずれか一方のアクセッサが実行中の動作を中断し、必要ならば邪魔にならない位置に待避したのちに、他方のアクセッサが動作を続行する必要がある。このように２台のアクセッサが互いに独立に動作できない状況を「干渉」とよぶ。この干渉を回避する方法を開発することが本章の主題である。

従来のライブラリ装置では、信頼性の向上を目的として２台のハンドリング機構を設置していた。しかし、性能向上のためにそれらを同時に稼動させることはなく、あくまでも、一方のアクセッサが稼動している間は他方がその邪魔にならない位置で待機する待機予備方式がとられていた。これは、次の理由による。従来、入出力処理に使用するテープをどのドライブ装置にマウントするかは、ホスト計算機のオペレーティング・システム（ＯＳ）が、装置種別あるいは装置間の負荷バランスを基準に決定する。テープの収納位置、ドライブ装置の設置位置などは基準としない。これは、テープ操作を人手で行うことを前提としているため、運搬ジョブ実行時の干渉を考慮する必要がなかったからである。ところが、ライブラリ装置の場合には、その構造上の制約により、テープの収納場所とドライブ装置との位置関係が干渉発生の重要な要因となる。従って、従来のようなドライブ装置割当方法を前提とした場合には、アクセッサ２台を並列動作させてもその効果が小さい。

この問題の解決策として、ＯＳにテープとドライブ装置の物理的位置に関する情報をもたせて、それに基づいてドライブ装置の割当を行うことは、ＯＳを大幅に改造することになり、しかも、ＯＳの汎用性を低下させるため得策ではない。そこで、ライブラリコントローラがテープとドライブ装置の位置情報を管理し、ホスト計算機の問い合わせにより、指定されたテープに対して最適なドライブ装置を選択する「問い合わせドライブ選択方式」を提案する。図 2.2 に従って説明する。

(1) ドライブ装置の問い合わせ：通常、ホスト計算機のＯＳは、計算機ジョブを実行する直前にそのジョブで使用する全てのテープに対してドライブ装置を割り当てる。このとき、本方式では、テープ番号と無作業状態の全ドライブ装置の番号をライブラリコントローラに対して通知して、それらドライブ装置のうちのどれを使用するべきかをコントローラに問い合わせる。

(2) ドライブ装置の選択：ライブラリコントローラは、ＯＳから示されたテープと、ドライブ装置群について、ライブラリ内の位置を調べる。そして、アクセッサ間の干渉が最も少なくなるようなドライブ装置を１つ選択してＯＳに知らせる。

(3) ドライブ装置の割当：ホスト計算機のＯＳは、上記の方法により、計算機ジョブで使用する全てのテープに対してドライブ装置を割り当て、ライブラリコントローラに対して、収納セルからドライブ装置までのテープの運搬を指示する。

(4) テープの運搬：ライブラリコントローラは、指示されたテープ運搬ジョブを、２台のアクセッサのうち最適なものに割り当てる。各アクセッサは、割り当てられたジョブを順次実行する。

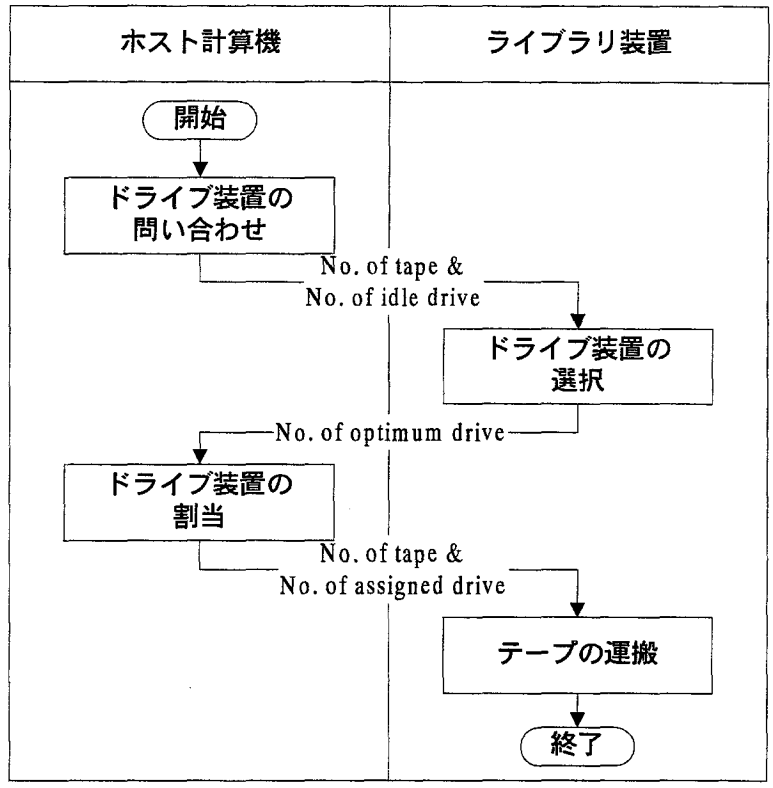


図 2.2　問い合わせドライブ選択方式

以上がテープをマウントする場合の手順である。デマウントの場合は、ドライブ装置が既に決まっているためその選択は不要であり、アクセッサの最適割当だけを行う。

この方法により、ホスト計算機のＯＳがライブラリ内におけるドライブ装置の物理的位置の情報をもっていなくても、干渉が最も少なくなるようなドライブ装置を選択することができる。これにより、２台のアクセッサを並列動作させ、その干渉を回避することが可能となる。すなわち、干渉回避問題は次のように定義することができる。

＜目的関数＞

テープ運搬ジョブの実行時間の総和を最小化するために、干渉を回避すること。

＜制約条件＞

a. ランダム性：テープマウント要求の発生時刻、テープの収納位置、テープ一巻あたりに記録／再生するデータ量は予め予測することができずランダムな分布をとるものとする。

b. マウント／デマウントの相違：テープをあるドライブ装置にマウントしたら、処理後、必ずそのドライブ装置からテープをデマウントして元の位置に戻すものとする。マウントジョブとデマウントジョブの相違は、テープ運搬の方向が、前者はセルからドライブ装置であるのに対して、後者はその逆であることだけで、その他の相違はないものとす

る。

c. アクセッサの待機点：各アクセッサは、一つの運搬ジョブの実行を終了したら、その終点で次のジョブが割り当てられるまで待機するものとする。

d. 計算機ジョブの実行性能：ホスト計算機のＯＳは、計算機ジョブの処理順序に基づいてテープのマウント／デマウント要求をだす。従って、テープ運搬ジョブを要求の発生順に処理しないことが、計算機ジョブの実行性能を低下させることがある。

＜操作変数＞

① 要求されたテープに割り当てるドライブ装置、

② 運搬ジョブを実行するアクセッサとその実行順序、

③ 干渉発生時の運搬ジョブの中断と実行の動作シーケンス。

2.4 干渉回避制御方式

2.4.1 基本的考え方

2.3 節で述べた操作変数のうち、①および②は干渉そのものを回避するための事前策である。これに対して、③は既に干渉が発生した時に極力その影響をなくすための事後処理である。明らかに、前者の方が後者よりも大きな効果を期待できるため、前者の干渉回避策に力点をおく。このように、ライブラリ装置の干渉回避問題を「要求されたテープの処理作業を、干渉を最小化する最適なドライブ装置とアクセッサに割り当てる問題」ととらえれば、これは一種の作業割当問題（ジョブスケジューリング問題）とみなすことができる。そして、作業の割り当て方と干渉の発生との関係をどのようにモデル化するかが、この問題を解決するうえでのポイントとなる。

一般に、作業割当問題とは、作業群をサーバ群により実行するとき、応答性能、処理性能などの評価関数を最適にするように、各サーバにおける作業の処理順序を決定する問題である。本問題は、作業の到着のしかたにより、静的問題と動的問題とに分類できる[48]。静的問題では、全ての作業がシステム内に同時に到着する。また、通常、作業の処理時間も予め確定している。この問題は、基本的には組み合わせ最適化問題であり、計算量の爆発を回避することを主目的として、従来から活発に研究が行われている[49]。これに対して、動的問題では、作業はシステム内に断続的に到着し、その到着時刻については統計的なことしかいえない。このため、動的問題は、組み合わせ最適化だけでなく、確率過程の最適制御問題でもある。それだけに、静的問題に比べて解決が困難であるといえよう[50,51]。これまでに、作業の到着過程がポアソン分布で、処理時間が指数分布であることを前提とし

て、先着順、後着順などの限定された割当方式を対象としてのみ解析が行われている[52]。

ライブラリ装置の運搬ジョブ割当問題は動的問題に属し、次の理由により従来の解析的方法が適用できない。

(1) ライブラリに到着する運搬ジョブの内容（始点と終点）が不確定で事前に予測できない。

(2) ジョブの長さ（運搬時間）は干渉の発生に大きく左右され、しかも、干渉が発生するか否かは、ジョブの割り当て方に依存する。

そこで、ジョブ内容が不確定のもとで干渉の発生を近似的に予測するヒューリスティックな方式を提案する。これを図 2.3 に従って説明する。

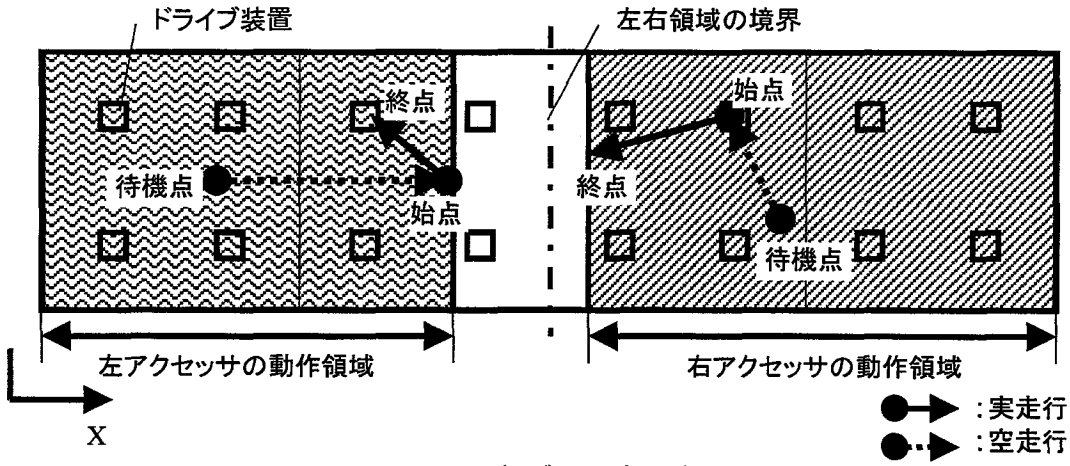


図 2.3　ジョブの干渉モデル

同時に実行する２つのジョブが干渉しないための必要条件は、アクセッサの待機点、および、ジョブの始点と終点によって決まる各アクセッサの動作領域が重複しないことである。ここで、左側のアクセッサが占有する動作領域とは、ジョブ実行開始時のアクセッサの待機点、ジョブの始点、終点のうちの最右点とライブラリの左端とにはさまれた領域である。右側のアクセッサの動作領域とは、待機点、始点、終点のうちの最左点とライブラリの右端とにはさまれた領域である。ところで、アクセッサの待機点はその直前に実行したジョブに依存するため、同時に実行しようとする２つのジョブだけから干渉の発生を予測することは、厳密には困難である。しかし、待機点は、アクセッサがジョブを実行開始した直後に離れる点であるため、その点の周辺にアクセッサが留まっている時間は無視し得る。そこで、ジョブの始点と終点だけを考慮して、次式により干渉が発生しないことを判定する。ここで、ライブラリ装置の左端を原点として、水平方向に右にのびるX座標軸を設定する。

$$\min(XS_r, XD_r) - \max(XS_l, XD_l) > L \qquad (2.1)$$

L：２台のアクセッサが接近できる限界距離

XS_r：右アクセッサが実行する作業の始点のＸ座標

XD_r：右アクセッサが実行する作業の終点のＸ座標

XS_l：左アクセッサが実行する作業の始点のＸ座標

XD_l：左アクセッサが実行する作業の終点のＸ座標

以下に、この条件式に基づいた干渉回避制御方法を詳細に述べる。

2.4.2 制御方式

本制御方式は、図 2.4 に示すように、最適ドライブ装置の選択、最適アクセッサへのジョブ割当、アクセッサの動作制御の３ステップで構成する。以下に、各ステップの詳細を説明する。

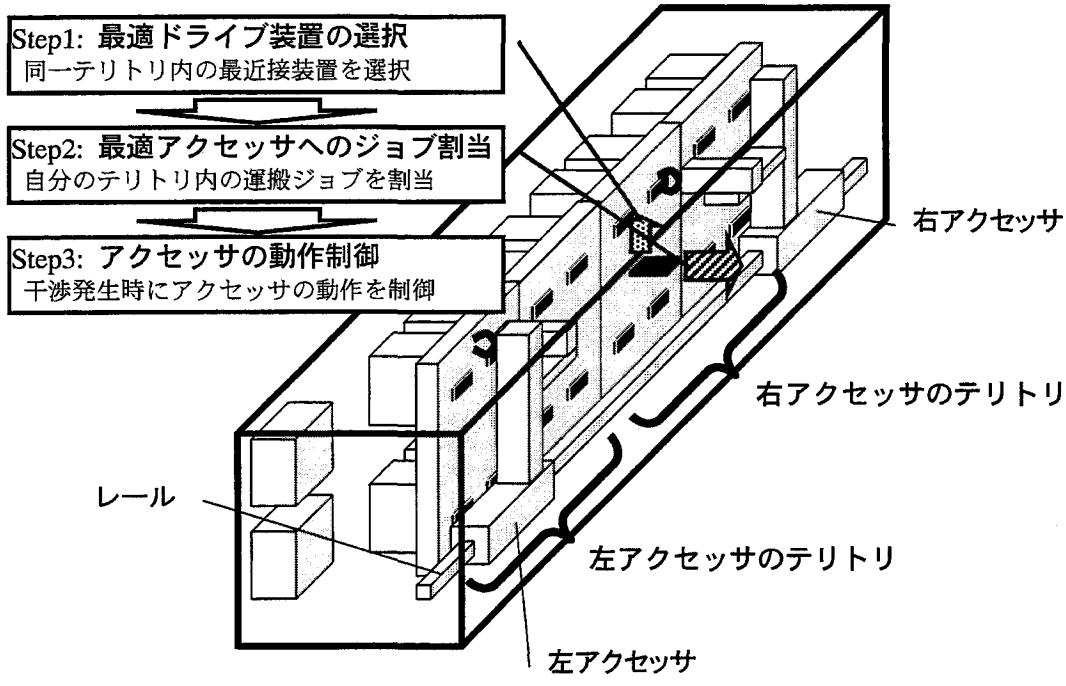


図 2.4　ライブラリ装置の制御方式

(1) 最適ドライブ装置の選択

要求されたテープにドライブ装置を割り当てる時点では、ジョブを実行するアクセッサとジョブの実行順序はまだ決まっていない。しかし、問題の前提条件により、ホスト計算機から要求されるテープはライブラリの全ての収納セルからランダムに選ばれるため、

ジョブスケジューリングを行えば、ライブラリの左半分に収納されているテープは左アクセッサが運搬し、右半分に収納されているテープは右アクセッサが運搬できる。すなわち、図 2.5 に示すように、ライブラリの収納セルを左右に二分し、各々を左アクセッサと右アクセッサのテリトリとするのである。左右のアクセッサが常に自分のテリトリ内で運搬ジョブを実行できれば、干渉は少なくなる。そのために、要求されたテープに対して、その収納位置と同一テリトリにあるドライブ装置を選択する。さらに、仮に干渉が発生してもそれに伴う待ち時間を最小化するために、テープ収納位置からなるべく近いドライブ装置を選択する。以上の議論により、ドライブ選択の基準は次の通りである。

i) 同一テリトリ内ドライブの選択：要求されたテープに対して、その収納位置と同一テリトリ内のドライブ装置を選択する。この際、なるべく左右のアクセッサの動作領域が接近しないように、ライブラリの左半分に収納されているテープに対しては、そのセルの位置よりも左側にあるドライブ装置を選択し、右半分に収納されているテープに対しては、そのセルの位置よりも右側にあるドライブ装置を選択する。この条件を満たすドライブ装置が一つもない場合には、残りのドライブ装置の中から選択する。

ii) 近接ドライブの選択：上記 i) の条件を満たすドライブ装置が複数個ある場合には、その中からテープの収納位置に最も近接した装置を選択する。また、条件を満たすドライブ装置がない場合も同様に最も近いものを選択する。

以上により、アクセッサ間の干渉発生を低減できるとともに、アクセッサの走行時間を短縮するという付随効果も得られる。

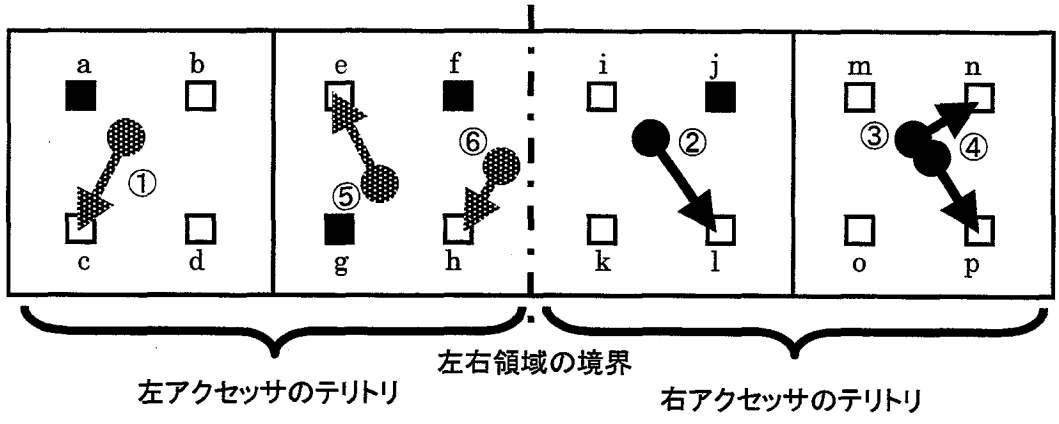


図 2.5　ドライブ装置選択の例

ドライブ装置選択の具体例を図 2.5 に示す。この図は、ホスト計算機から要求されたテープが図の●印で示される位置に収納され、□印のドライブ装置が未使用、■印のドライ

ブ装置が使用中であると仮定した場合、各テープに対してどのドライブ装置を選択するのかを示したものである。例えば、図 2.5 におけるテープ⑤は左アクセッサのテリトリ内に収納されているため、収納セルの左側にあり最も近い未使用のドライブ装置 e を選択する。

(2) 最適アクセッサへのジョブ割当

要求されたテープの運搬作業は、アクセッサに割り当てられるべきジョブとして、実行待ち行列に格納される。この中から、左右のアクセッサが互いに干渉する可能性が低くなるようにジョブを割り当てることが必要となる。ただし、2.3 節で述べたように、ホスト計算機ＯＳの定めたジョブの実行順序を、極力遵守することが必要である。

前述したドライブ装置の最適選択を行うことにより、各ジョブは、左右の各テリトリ内で実行できる干渉が発生しにくいものとなっている。このため、常に左右のアクセッサが、各々、自分のテリトリ内のジョブを実行すれば、アクセッサの動作領域は重複せず干渉は発生しない。ところが、次のような場合に干渉が発生する可能性がある。

a. 境界線付近でのジョブの実行：左右のアクセッサが、各々、自分のテリトリ内のジョブを実行したとしても、２台が同時に境界線付近でジョブを実行する場合には動作領域が重複することがある。

b. 応援ジョブの実行：ホスト計算機から要求されるテープの収納位置はランダムであるため、左半分に収納されているテープ、あるいは、右半分に収納されているテープばかりが連続して要求される場合がある。このような場合、一方のアクセッサが他方の応援を行うために境界線を越えることがある。

最適アクセッサへのジョブ割当では、以上２つの原因で発生する干渉を、次の方法により回避する。

i) 接近ジョブ実行の回避：２台のアクセッサが互いに接近した位置でジョブを実行しないように、(2.1)式の条件を満たすジョブを優先して割り当てる。この条件を満たすジョブが複数個ある場合には、ＯＳから指定された実行順序を遵守するために、マウント／デマウント要求の到着順に割り当てる。また、特定のジョブが後発のジョブに追い越され長時間実行を待たされる、いわゆる、沈み込みを防ぐために、予め定められた回数以上に先を越されたジョブは、その次の割当時点にて最優先で割り当てることにする。

ii) 非効率ジョブの同時実行の禁止：干渉に伴う待ちあるいは待避動作が多発することによって、２つのジョブを並列に実行するよりもシーケンシャルに実行するほうが効率がよい場合がある。すなわち、割り当てようとしているジョブの始点と終点の両方と

も、ジョブ実行中のアクセッサの動作領域内にある場合である。従って、この場合にはアクセッサにジョブを割り当てず、実行中のジョブが終了した後でその保留されたジョブを割り当てる。

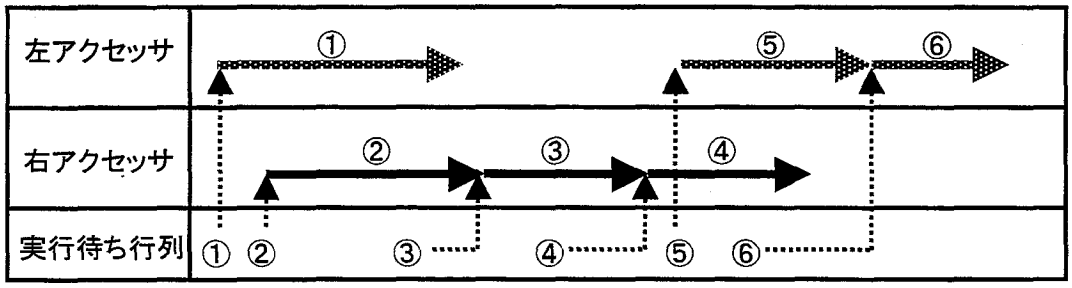


図 2.6　アクセッサ割当の例

アクセッサ割当の具体例を図 2.6 に示す。この図は、図 2.5 においてドライブ装置を選択した結果生じた運搬ジョブを、左右どちらのアクセッサに割り当てるかを示したものである。図の横軸は時間の流れを示す。縦軸は、各々、左アクセッサの実行するジョブの割当、右アクセッサの実行するジョブの割当、実行待ち行列内のジョブを示す。例えば、ジョブ①が待ち行列に入った時点を考える。左右のアクセッサはともに無作業状態であるため、どちらのアクセッサにも割当可能であるが、ジョブ①は左アクセッサのテリトリ内のジョブであるため、左アクセッサに割り当てる。次に、ジョブ③が待ち行列に入った時点を考える。左右のアクセッサはともに作業中であるため、ジョブ③は待ち状態となる。初めに左アクセッサの作業が終了するが、ジョブ③は右アクセッサのテリトリ内のジョブであるため割当は行わない。そして、右アクセッサの作業が終了した時点で割当を行う。

(3) アクセッサの動作制御

以上に述べた 2 階層の最適化を行うことにより、干渉発生の頻度は非常に低くなるが、それをゼロにすることはできない。そこで、干渉が発生した場合でもその待ち時間が極力短くなるように、アクセッサの待ちと待避動作の実行シーケンスを決定する。ただし、アクセッサの動作を細分して停止と起動を繰り返しても、それに伴うオーバヘッドによりかえって性能が低下するため、次のような単純な方法で制御する。

a. アクセッサのジョブを、空走行動作（待機点から始点までテープを取りに行く動作）と実走行動作（テープを始点から終点まで運搬する動作）とに分け、各動作の開始時にだけ干渉が発生するか否かのチェックを行う。

b. 各アクセッサは、自分が動作を開始する前に、最悪の場合に相手とどのくらい接近するかを見積もる。その結果、干渉が発生する場合は相手の動作終了を待つ。この時に、待機点が相手の動作領域内にある場合は、領域外まで移動してそこで待機する（これを待避とよぶ)。そうでない場合は動作を開始する。このように待避動作を行うことにより、アクセッサのデッドロックを防止することができる。

具体的には、次式により干渉が発生しないことを判定する。

$$\min(XP_r, XD_r) - \max(XP_l, XD_l) > L \qquad (2.2)$$

L：２台のアクセッサが接近できる限界距離

XP_r：右アクセッサの現在位置のＸ座標

XD_r：右アクセッサが実行中あるいは実行しようとしている動作の目標位置のＸ座標

XP_l：左アクセッサの現在位置のＸ座標

XD_l：左アクセッサが実行中あるいは実行しようとしている動作の目標位置のＸ座標

2.5 評価

2.5.1 評価条件

以上に述べた制御方式の効果をシミュレーションにより評価する。評価条件は表 2.1 に示す通りである。以下にその概要を説明する。

(1) 利用形態

ライブラリ装置の利用形態を、「ホスト計算機からの要求がいつ発生し、各要求において何巻のテープに、どのくらいのデータ量をリード／ライトするか」、すなわち、入出力要求発生間隔、同時要求テープ巻数、テープ１巻あたりのリード／ライトデータ量という３項目のデータによりモデル化する。この３項目のデータが具体的にどのような値をとるかは、ライブラリ装置をどういう目的で利用するのかによって種々に異なる。ライブラリ装置の利用目的として典型的なものは、バッチジョブの自動実行とデータベースの自動バックアップである。バッチジョブの自動実行に利用する場合は、入出力要求は不定期に発生し、各要求で用いるテープの巻数、リード／ライトデータ量もランダムな値をとる。これに対して、データベースの自動バックアップに利用する場合は、入出力要求は連続的に発生し、各要求ではテープを１巻ずつ用いてテープエンドまでデータをライトする。前者の利用形態は後者に比べてテープの運搬ジョブを頻繁に必要とするため、干渉が発生しやすい。そこで、悪条件のもとでの効果を調べるために、前者の形態を想定して評価を行う。具体的には次の条件で行う。

・入出力要求の発生はポアソン過程に従う。

・１回の入出力処理では１巻から５巻までのテープをランダムな数だけ要求する。

・テープにリード／ライトするデータ量は０ＭＢから 200ＭＢ（テープの最大容量）までの値をランダムにとる。

・要求されるテープはライブラリの全セルにランダムに収納されているものとする。

(2) ハード諸元

ライブラリ装置のタイプは、テープ 6500 巻収納、ドライブ装置 32 台搭載の最大構成タイプとする。アクセッサの走行特性は表 2.1 に示す通りであり、起動時の加速と停止時の減速は線形的に行うものとする。ドライブ装置の動作特性も表 2.1 に示す。テープ装着および取外し時間は一定であり、リード／ライト時間および巻戻し時間は処理データ量に比例するものとする。

表 2.1　評価条件

#	条件項目		値
1	利用形態	入出力要求発生間隔	平均λ[回/分]の指数分布
		同時要求テープ巻数	1-5[巻/回]の範囲で一様分布
		１巻あたり R/W データ量	0-200[MB] の範囲で一様分布
		テープ収納位置	全棚に一様分布
2	ハード諸元	ライブラリ製品タイプ	最大構成タイプ 幅：16[m]、奥行：1.4[m]、高さ：1.9[m] ドライブ装置 32 台 ＭＴコントローラ：4 台
		アクセッサ性能	走行速度：（水平）1.5[m/秒] （垂直）1.0[m/秒] 加速時間：　0.8[秒] 減速時間：　1.4[秒]
		ドライブ装置性能	テープ装着時間：10[秒] テープ取外時間：10[秒] データ転送時間：2.7[秒] テープ巻戻時間：Max. 40[秒]

(3) 比較対象

提案した制御方式を次の三つの方式と比較する。

a. ジョブ割当単独方式：問い合わせドライブ決定方式により、最適なドライブ装置を選択する効果をみるために、これを行わずにアクセッサへの最適ジョブ割当だけを

行う方式を評価する。

b. 待機予備方式：従来方式と比較するために、アクセッサ１台でサービスする待機予備方式を評価する。

c. ＦＩＦＯ方式：アクセッサ２台を最悪な制御方法で並列動作させる場合の性能をみるために、ドライブ装置をランダムに選択し、アクセッサに対してジョブを到着順に割り当てるＦＩＦＯ（First In First Out）方式を評価する。

2.5.2 評価結果

図 2.7 に応答時間、図 2.8 にテープ運搬ジョブの処理時間の評価結果を示す。グラフの横軸はマウント要求の到着率を示す。ここで、マウント要求の到着率とは、計算機ジョブからの入出力要求の平均発生率と、一回あたりの入出力要求で使用する平均テープ巻数との積である。また、各マウント要求に対応して、デマウント要求が、テープ装着、リード／ライト、テープ取外を行った後に発生するものとする。

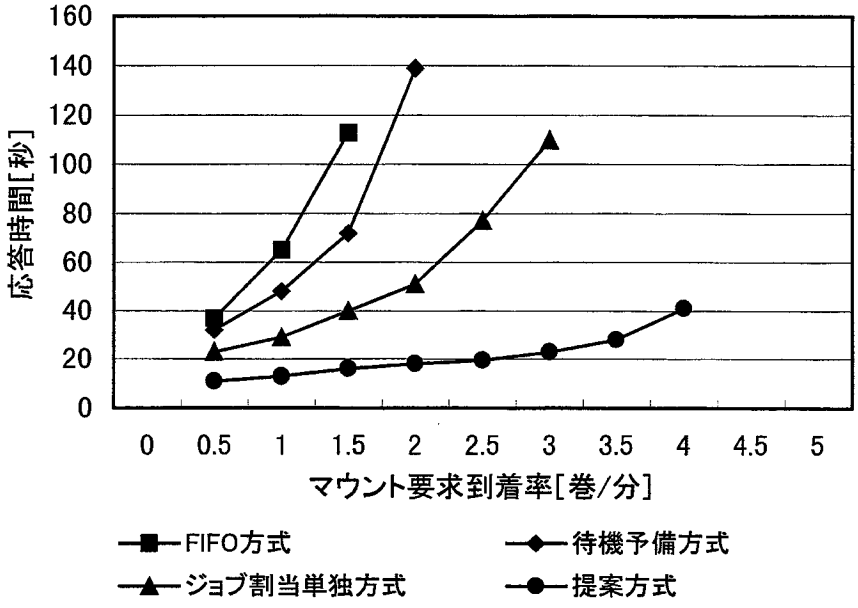


図 2.7　応答時間のシミュレーション結果

図 2.7 を見ると、提案方式は他の三つの方式に比べて応答時間を大幅に低減できることがわかる。提案方式では、ホスト計算機からのマウント要求の到着率が 4.0[巻／分]までは、要求されたテープを平均 40 秒以下でマウント完了できる。これに対して、待機予備方式およびＦＩＦＯ方式では、マウント要求の到着率が 1.0[巻／分]付近の低いレベルで

も 40 秒以上の応答時間がかかり、到着率が 2.0[巻／分]を越えると、もはや実用的な時間では処理不可能となってしまう。ジョブ割当単独方式では、待機予備方式およびＦＩＦＯ方式に比べて性能の向上がみられるが、せいぜい、アクセッサ 1.5 台分の処理性能である 3.0[巻／分]しか処理できない。これに対して、提案方式では、２台のアクセッサがほとんど干渉することなく並行動作することができ、4.0[巻／分]まで処理できる。また、図に示すように、ＦＩＦＯ方式は干渉に伴う待ちおよび待避動作が多発するため、アクセッサ１台でサービスする待機予備方式よりも性能が悪くなる。

このように提案方式の効果が顕著なのは、図 2.8 に示すように、マウント要求の到着率が増加しても干渉をほぼゼロに抑え、ジョブ処理時間を 10 秒以下に保てるからである。これに対して、他の方式では、マウント要求の到着率が低い時点でもジョブ処理時間が提案方式の 1.5 倍以上にもなる。そして、到着率が増加するのにともなって着実に増加してしまう。以上により、干渉回避のためのドライブ装置選択およびアクセッサ割当が効果的に働いていることがわかる。

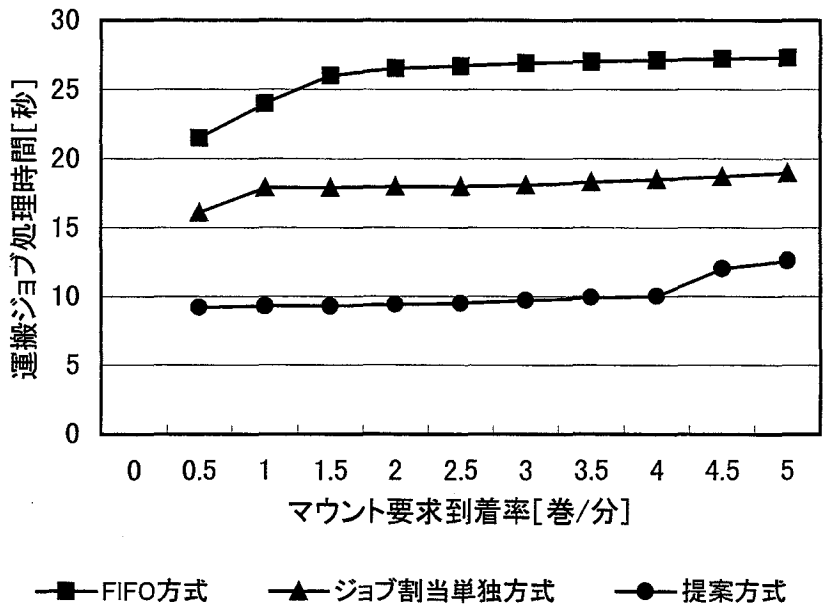


図 2.8　運搬ジョブ処理時間のシミュレーション結果

2.6　結言

本章では、計算機入出力装置の一種であるカートリッジ磁気テープライブラリ装置の共

有リソースの競合回避を題材として、段階的にヒューリスティックスを適用するジョブ管理方式の開発方法を検討した。まず、テープ運搬ジョブの割当問題を、①最適ドライブ装置の選択、②最適アクセッサへのジョブ割当、の２つのサブ問題に分割した。そして、最適ドライブ選択において左アクセッサのテリトリ（ライブラリの左半分領域）または右アクセッサのテリトリ（ライブラリの右半分領域）だけで実施できる運搬ジョブを生成し、最適アクセッサ割当において左右のアクセッサに自分のテリトリのジョブを割り当てる方針をとった。これにより、２つのサブ問題が相互作用の少ない単純なものとなり、ヒューリスティックスを適用して解くことができるようになった。また、問題を分割しない方式に比べて、アクセッサ間の干渉を大幅に低減でき、応答性能とスループットを向上した。

本章で提案した制御方式は、左右２台のアクセッサを対象としたものである。しかし、各アクセッサに対してテリトリを設定してドライブ装置選択とアクセッサ割当を行う方式は、アクセッサの台数がｎ台（ｎ＞２）となっても適用可能である。今後、より多くのアクセッサを必要とする大規模なライブラリ装置への適用を検討する。

本制御方式は（株）日立製作所のカートリッジ磁気テープライブラリ装置に搭載され、多くの顧客サイトで稼働中である[53]。

第3章　ジョブの干渉回避スケジューリング方式の確率ペトリネットを適用した性能解析

3.1　緒言

本章では、共有リソースの競合によるジョブの干渉を回避するためのヒューリスティックなスケジューリング方式を対象として、その評価方法を検討する[54,55,56]。

分散処理システムでは、共有リソースの競合によるジョブ間の干渉がシステム性能低下の大きな要因となることが多い。そして、リソース競合を解消するために、ジョブスケジューリング方式をいかに最適なものとするかが課題となる。このため、システム計画、設計の段階で、考案されたスケジューリング方式の代替案を比較評価できることが必要となる。

ジョブがランダムに到着し、その処理時間が一定でなく、複数のリソースにより構成されるシステムの性能評価は、従来から、待ち行列ネットワークを中心とした確率理論を適用して行われていた[22,23,57,58]。しかし、ネットワークの各ノードにおいてリソースの割当制御（すなわちスケジューリング）を行う場合、システムの状態によりノードのルーティングを変更する場合など、実用上頻繁に現れる条件下において、解析的な解法が開発されていない。また、モデル自体に、ジョブ間の同期、リソースの排他などを表現するための機能がない。このため、その適用は狭い範囲に限定される。本章で対象としているヒューリスティックなスケジューリング方式の評価を、従来の待ち行列ネットモデルで解析することは困難である。一方、離散シミュレーションによる評価は、汎用言語やシミュレーション言語を用いたコーディングに基づくため、そのモデル化範囲は広い[59,60,61]。しかし、モデル作成、プログラム開発、シミュレーション実施、結果の解釈等に膨大な工数と経験を必要とするという問題点がある[61]。

上述の待ち行列ネットモデルの問題点を解決しようとするものが、1982 年にテキサス大学の M. K. Molloy によって提案された確率ペトリネット（SPN: Stochastic Petri Nets）

である[62,63,64]。ＳＰＮは、従来のペトリネットの全てのトランジションに指数分布の発火確率を設定し、条件が成立した後に瞬時に発火するのでなく、与えられた指数分布確率に従って遅延して発火するようにしたものである。ＳＰＮの特徴は次の通りである。

(1) 強力なモデル化能力：ペトリネットのモデル化能力は高く、それはチューリング機械と等価である[65,66,67]。ＳＰＮは、この能力を引き継ぐとともに、確率過程をモデル化できる能力をもっている。

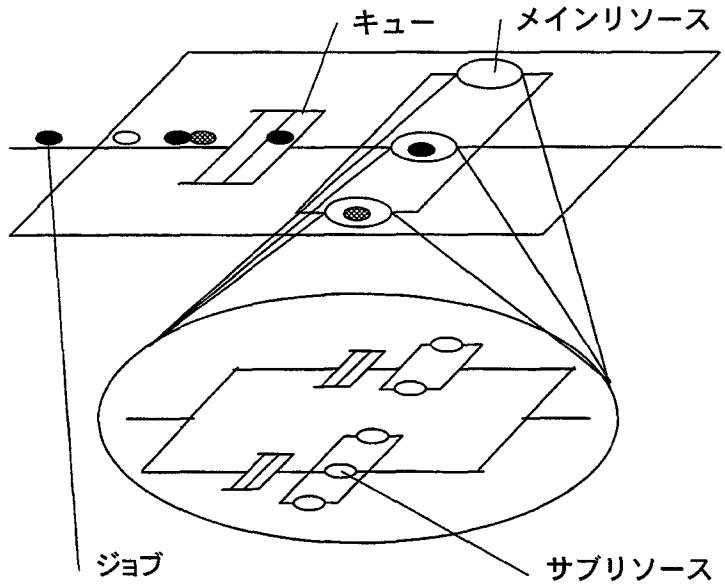
(2) マルコフモデルと同等な解析能力：マルコフモデルを適用するためには、システム状態を漏れなく洗い出すこと、および、それら状態間の遷移関係と遷移確率を見いだすことが課題となる[2,50]。ＳＰＮでシステムをモデル化すれば、そのモデルから機械的な操作でマルコフモデルを作成することができる[62]。

ＳＰＮによれば、ジョブの到着状況、リソースの割当状況、および、それらに基づくジョブの流れをモデル化することができ、本章で対象とするようなスケジューリング方式のモデル化も理論的には可能である。すなわち、ペトリネットのプレースにより条件を、トランジションにより事象を記述することにより、「もし～という条件が成立すれば、～という事象が発生する」というプリミテイブな論理を表すことができる。これを組み合わすことにより、複雑なスケジューリングの論理を構成する。しかし、このような方法でスケジューリング方式をモデル化して解析する場合、障害となるのは複雑性の問題である。ペトリネットの要素であるプレース、トランジションなどの個々の記述能力は、プログラミング言語に例えればアセンブラ程度であるため、上記の方法で作成するモデルは膨大なプレースとトランジションを伴う。このため、モデル化自体が難しいとともに、仮にモデル化できたとしてもその解析は事実上不可能である。スケジューリング方式の解析にＳＰＮを適用するためには、モデルの規模をいかに縮小するかが課題となる。

本章では、共有リソースの競合によりジョブの干渉が発生するシステムを対象として、その干渉回避スケジューリング方式の解析方式を提案する。提案方式では、まず、着目しているスケジューリング方式がシステム状態（ジョブの到着状況、リソースの割当状況など）に応じてどのような処理性能を実現するかという性能関数を導く。そして、この性能関数を確率ペトリネットモデルに組み込むことによりモデルの規模を縮小する。さらに、提案方式を磁気テープライブラリ装置の制御方式の性能解析に適用して、その有効性を示す。

3.2 スケジューリング方式性能解析問題

ジョブの実行と制御をつかさどるメインリソースが複数あり、それらがジョブの処理内容に応じて各種のサブリソースを使用して並列に処理を進めるシステムを考える（これを分散処理システムとよぶ、図 3.1 を参照）。例えば、マルチプロセッサシステムでは、メインリソースはＣＰＵであり、サブリソースは共通テーブル、バッファなどである。ジョブの干渉とは、複数個のジョブが並列処理中に同時に同一のサブリソースにアクセスしたため、そのうちの１つのジョブだけが処理を続行でき、他のジョブはそのサブリソースが空くまで処理を中断して待たなければならない状況をいう。



到着時間間隔、処理時間は指数分布に従う。
グループごとに異なるサブリソースを使用。

図 3.1　分散処理システム

メインリソースは、ジョブを割り当てられてからそれが終了するまで他のジョブを処理しないものとする。従って、一つのジョブの実行によりメインリソースが占有される時間は、メインリソースが単独で処理を行う時間、サブリソースを使って処理を行う時間、サブリソースの競合待ち時間、の３つの時間の和である。干渉回避スケジューリングとは、サブリソースの競合を回避してメインリソースの占有時間を最小化することを目的としている。そして、そのためにメインリソースの実行待ち行列内にある各種ジョブの中から最適なものをメインリソースに割り当てる。

以下に、ジョブ、メインリソース、サブリソース、干渉回避スケジューリングを定義する。

a. ジョブ

・対象とするシステムはオープン系であり、ジョブは外部から到着してシステム内で処理された後に再び外部に出ていく。

・ジョブはその処理内容により複数個のグループに分類することができ、各グループごとに使用するサブリソースのタイプと数が予め決まっている。

・ジョブの到着時間間隔と処理時間の分布は指数分布に従う。また、グループごとのジョブ数の割合は事前にわかっている。

b. メインリソース

・メインリソースは複数個あり、これらにより、与えられたジョブを並列に実行する。また、全てのジョブはどのメインリソースでも実行できる。

c. サブリソース

・サブリソースは複数のタイプに分けることができ、各タイプごとに複数個ある。各ジョブはその処理内容に応じたタイプのサブリソースを使用する。

・１つのサブリソースは、同時には１つのジョブだけが使用できる。従って、ある１つのサブリソースに対して複数のジョブから同時にアクセス要求が発生した場合、そのうちのただ１つのジョブだけがアクセス権を獲得できる。

d. 干渉回避スケジューリング方式

本章で対象とするジョブスケジューリングを次のように定義する。

＜目的関数＞

分散処理システムにおけるジョブの滞留時間を低減し、スループットを向上するために、サブリソースの競合を起因とするジョブ間の干渉を回避すること。

＜制約条件＞

メインリソースの実行待ち行列内のジョブ数とその所属グループ、メインリソースで処理中のジョブ数とその所属グループ、および、使用中のサブリソースの数とそのタイプを観測可能とする。

＜決定変数＞

メインリソースにおけるジョブの実行順序。

本章では、以上に述べた分散処理システムのスケジューリング方式の代替案が与えられた時に、それらの代替案の性能解析方法を考える。

3.3 確率ペトリネット適用解析方式

3.3.1 考え方

スケジューリング方式の性能解析では、ジョブをどのくらいの時間で処理できるかという性能面だけをモデル化すればよく、スケジューリング論理を忠実に模擬するようなモデルを作る必要はない。そこで、本章で提案する解析方式では、スケジューリング方式の各代替案ごとにシステム状態とその時のジョブ処理性能との関係を導く。これを性能関数とよぶ。スケジューリング方式をこの性能関数によりモデル化することで、分散処理システム全体のＳＰＮモデルを解析可能な規模のものとすることができる。すなわち、次の２段階の方法で分散処理システムの性能評価モデルを作成する。

(1) スケジューリング方式を性能関数としてモデル化する：図 3.1 で定義した分散処理システムにおいて、ジョブの到着状況、ジョブの処理状況、リソースの割当状況などのシステム状態が与えられた時、各スケジューリング方式によればジョブをどのくらいの性能で処理できるかという性能関数を求める。この性能関数が、各スケジューリング方式の性能解析モデルとなる。これにより、スケジューリング論理を明示的に記述する必要がなくなる。

(2) 性能関数を利用してＳＰＮモデルを作成する：分散処理システムをＳＰＮでモデル化する。この際、(1)で求めた性能関数を、ジョブの処理終了を表すトランジションの発火率としてモデルに組み込む。

このようにして作成したＳＰＮモデルは機械的操作によりマルコフモデルに変換することができ、マルコフモデルの解析手法を適用して解析することができる。

以上に述べた方式を、次に詳細に説明する。

3.3.2 解析方式

(1) スケジューリング方式のモデル化

3.2 節で述べた問題の前提により、あるスケジューリング方式をとった場合、サブリソースの競合が発生するか否かは次の３種類の状態ＳＱ、ＳＭ、ＳＲにより決まる。そして、サブシステム全体の状態Ｓは、これらの直積ＳＱ×ＳＭ×ＳＲにより記述できる。

・メインリソースの待ち行列におけるジョブの到着状況：

$SQ = \{Q_1, Q_2, \cdots Q_x\}$

Q_j：ジョブグループ j (j=1,2,…,x)に属するジョブのうち、メインリソースの待ち行列内にあるジョブの数。

・メインリソースの割当状況：

$SM = \{M_1, M_2, \cdots, M_x\}$

M_j：ジョブグループ j (j=1,2,…,x)に属するジョブのうち、メインリソースで処理中のジョブの数。

・サブリソースの割当状況：

$SR = \{R_1, R_2, \cdots, R_y\}$

R_k：リソースタイプ k (k=1,2,…,y)に属するサブリソースのうち、使用状態にあるサブリソースの数。

状態 $s_i \in S$ においてサブリソースの競合が発生すると、1ジョブあたりのメインリソースの占有時間はサブリソースの競合待ち時間だけ増加する。従って、状態 $s_i \in S$ におけるメインリソースの占有時間 TP_i は次式で表せる。

$$TP_i = TM + TR + TW_i \qquad (3.1)$$

TM ：メインリソース単独処理の平均時間。

TR ：サブリソース使用処理の平均時間。

TW_i：状態 i におけるサブリソースの競合待ち時間。

ここで、TW_i は上記の3種類の状態SQ、SM、SRによって決まる s_i に応じて変化する。

ところが、上記のようにジョブグループとリソースタイプを区別するように状態を設定した場合、状態数はそれらの組み合わせにより指数関数的に増大してしまう。そこで、以下に状態数を削減する方法を述べる。まず、前節で述べたように、最終的に解析したいのはジョブ、リソースのタイプの区別をしないシステム全体の処理性能であるから、ジョブ、リソースの個数だけを考慮してモデルを作成する。すなわち、サブシステムの状態を次の3つの要素により記述する。

・Q：メインリソースの待ち行列内にあるジョブの数。

・M：メインリソースで処理中のジョブの数。

・R：使用状態にあるサブリソースの数。

問題の前提により、ジョブグループごとのジョブ数の割合は与えられているから、あるジョブがジョブグループ j (j=1,2,…,x)に属する確率は予めわかる。また、ジョブグループごとに使用するリソースのタイプと数が与えられているから、使用状態にあるサブリソー

スがリソースタイプk(k=1,2,…,y)に属する確率も予めわかる。従って、(Q, M, R)をもとに上記の(SQ, SM, SR)を確率的に予測することができる。さらに、同様の理由により、使用中のサブリソースの数Rは、メインリソースで処理中のジョブ数Mにより確率的に予測できる。

以上により、サブシステムの状態$s_i \in S$は(Q, M)により記述できる。従って、上記(3.1)式のTP_iを、待ち行列で実行待ち中のジョブの数Qと、メインリソースで処理中のジョブの数Mに依存する関数で次式のように記述することができる。

$$TP_i = f\ (Q,\ M) \qquad (3.2)$$

そして、あるスケジューリング方式をとった場合の単位時間あたりのジョブ処理数、すなわち、性能関数は、メインリソース占有時間の逆数$(TP_i)^{-1}$となる。図3.2は性能関数のイメージを示したものであり、メインリソースの待ち行列中のジョブ数Qと処理中のジョブ数Mに応じて、単位時間あたりの処理数$(TP_i)^{-1}$の値が決まる。

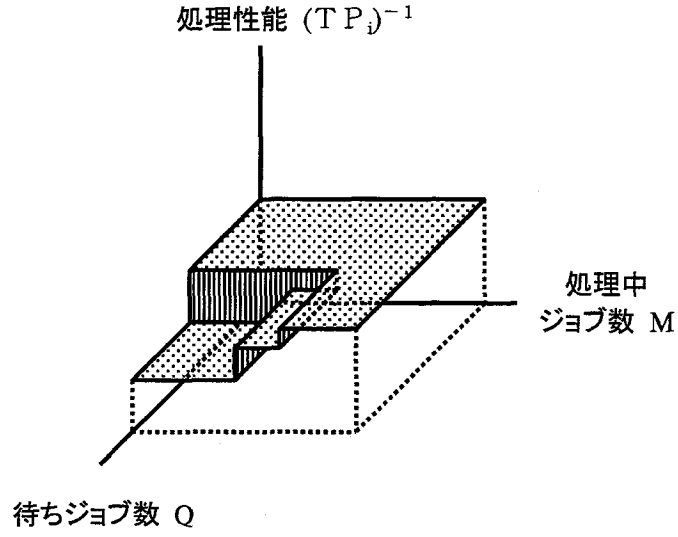


図3.2 性能関数のイメージ

(2) SPNモデルの作成

上記(1)により導いた性能関数を使って、図3.3に示すように分散処理システムのSPNモデルを作成する。本モデルでは、待ち行列で実行待ち中のジョブをプレースp_0で、メインリソースで処理中のジョブをプレースp_1で表す。そして、p_0のジョブ数をm_0、p_1のジョブ数をm_1とする。また、p_2、p_3は、各々、p_0、p_1の容量を制限するための相補的なプレースであり、待ち行列内の空き席の個数と空き状態のメインリソースの個数を示す。システム外部からのジョブの到着はトランジションt_0で表し、λ_0がその到着

率を示す。ジョブの処理終了は t_2 で表し、λ_1 がジョブの性能関数である$(TP_i)^{-1}$を示す。また、t_1はメインリソースへのジョブの割当を表す。

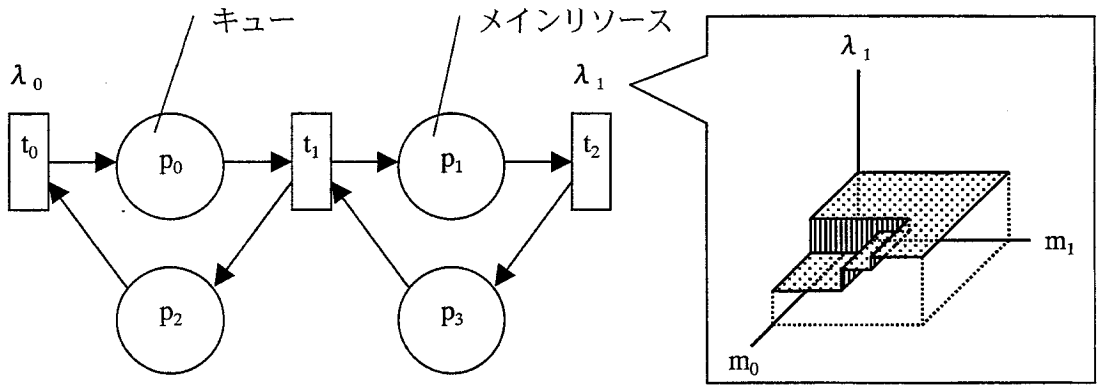


図 3.3　分散処理システムの SPN モデル

上記の(1)、(2)により作成したＳＰＮモデルを次の方法で解析する。すなわち、まず、このＳＰＮモデルをもとに分散処理システムの状態遷移図を作成する。これは、ペトリネットの可到達グラフ作成アルゴリズムを適用して、機械的操作により行うことができる。次に、マルコフモデルの解析手法に従って、この状態遷移図から個々の状態の存在確率を求める。そして、Little の公式を適用して、ジョブのターンアラウンドとスループットを算出する[22,62]。

3.4　磁気テープライブラリへの適用

本節では、3.3 節で提案した解析方式を磁気テープライブラリ装置の制御方式の性能評価に適用し、その有効性を検討する。

3.4.1　制御方式の解析

第２章で述べたように、磁気テープライブラリ装置では、運搬ジョブの干渉はレールの競合により発生する。このため、２台のアクセッサをメインリソースに、レール上に設定した左右２つのテリトリをサブリソースに対応づけることにより提案方式を適用する。システムに到着した運搬ジョブは、２台のアクセッサのいずれかにより実行する。この際、実行する運搬ジョブの始点と終点に応じて、左、右、あるいは、それら両方のテリトリを占有する。前章で述べたＱ、Ｍ、Ｒは、各々、実行待ちジョブ数、アクセッサで処理中の

ジョブ数、占有されているテリトリの数に対応する。ただし、到着したジョブがどのテリトリを使用するかは、どのテリトリのドライブ装置が空いているかにより決まる。このため、到着したジョブがどのジョブグループに属するかの確率は、処理中のドライブ装置台数の関数となる。以下に具体的な解析方法を述べる。

(1) 制御方式のモデル化

ジョブグループを次のように定義し、到着したジョブがそのジョブグループに属ずる確率を算出する。

・Lグループ：左テリトリだけで実行するジョブのグループ

・Rグループ：右テリトリだけで実行するジョブのグループ

・Wグループ：左右のテリトリにまたがって実行するジョブのグループ

まず、ドライブ装置の最適選択制御を行う場合について、各グループに属するジョブ数の割合を算出する。N_d 台のドライブ装置のうち m_2 台が処理中である時にテープの入出力要求が発生すると、残りの（N_d-m_2）台の中からドライブ装置を選択する必要がある。この際、（N_d-m_2）台のドライブ装置全てが要求されたテープと異なるテリトリにある場合には、ジョブはWグループになる。前章で述べたように、ライブラリ装置は左右対称の構造であるため、各ドライブ装置が左右のテリトリに一様に設置されていると仮定できる。従って、処理中のドライブ装置が全体の半数を越えた時、すなわち、$m_2/N_d > 0.5$ の時にこのような状況が発生する可能性がある。そして、任意のドライブ装置が要求されたテープと異なるテリトリにある確率は 0.5 であるから、（N_d-m_2）台全てが異なるテリトリにある確率は $0.5^{N_d-m_2}$ となる。それ以外の場合、すなわち、$m_2/N_d \leqq 0.5$ には、ジョブは必ずLあるいはRグループになる。以上により、ジョブがLあるいはRグループに属する確率ＰＤは次式で表すことができる。また、ジョブがWグループに属する確率は（１－ＰＤ）となる。

$$PD = \begin{cases} 1-0.5^{N_d-m_2} & (m_2/N_d > 0.5) \\ 1 & (m_2/N_d \leqq 0.5) \end{cases} \qquad (3.3)$$

これに対して、ドライブ装置をランダムに選択した場合には、確率ＰＤは処理中のドライブ装置の台数に依存せず一様に 0.5 となる

以上の前提のもとに、アクセッサへのジョブ割当方式の性能関数を求める。まず、最適ジョブ割当を行う場合を考える。ジョブの干渉を回避できるのは、待ち行列内にＬあるいはＲグループに属するジョブが少なくとも一つあり、しかも、そのテリトリを担当するアクセッサが他のジョブを実行中でない場合である。左右のアクセッサが両方とも無作業中である場合には、待ち行列内にＬあるいはＲグループに属するジョブが少なくとも一つあ

ればジョブの干渉は発生しない。これに対して、いずれか一方のアクセッサが作業中である場合には、作業中のアクセッサが自分の担当テリトリのジョブを実行中であり、しかも、待ち行列内に割当対象としているアクセッサのテリトリのジョブが少なくとも一つ存在することが必要である。待ち行列内にm_0個のジョブがある場合に、その中の任意のジョブが着目しているアクセッサのテリトリのものである確率は 0.5・ＰＤであり、そうでない確率は（１－0.5・ＰＤ）である。従って、m_0個のジョブのうち少なくとも一つが着目しているアクセッサの担当テリトリのものである確率ＰＴは次式により計算できる。

$$PT = 1 - (1 - 0.5 \cdot PD)^{m_0} \qquad (3.4)$$

作業中のアクセッサが自分のテリトリのジョブを実行している確率もこのＰＴと同じであると見なせるため、２台のアクセッサが両方とも稼働中の場合、すなわち、$m_1 > 1$の時、それらが干渉しない確率はPT^2となる。１台だけ稼動している場合、干渉は発生しない。従って、２台のアクセッサが干渉しない確率ＰＡは次式で表すことができる。

$$PA = \begin{cases} PT^2 & (m_1 > 1) \\ 1 & (m_1 \leqq 1) \end{cases} \qquad (3.5)$$

これに対して、先入れ先出しによりジョブを割り当てる場合、ＰＴの値は待ち行列の状態に依存せず一様に 0.5・ＰＤとなる。この場合も同様に、２台のアクセッサが干渉しない確率ＰＡは、(3.5)式により求めることができる。

さて、干渉が発生しない場合の平均運搬時間をt_0とし、干渉が発生した場合の平均運搬時間を$\alpha \cdot t_0$とすると、アクセッサの平均運搬時間ＴＰは次式で表せる。

$$TP = t_0 \cdot PA + \alpha \cdot t_0 \cdot (1 - PA) \qquad (3.6)$$

ここで、αは問題に応じて決まるパラメータである。今回の解析では、干渉発生時には一方の処理が他方の処理開始から終了まで常に待たされるという最悪の場合を想定して、$\alpha = 2$とする。アクセッサの性能関数は(3.6)式で求めたＴＰの逆数$(TP)^{-1}$となる。

以上により、アクセッサの性能関数を、待ち行列内のジョブ数m_0、稼働中のアクセッサ台数m_1、処理中のドライブ装置台数m_2の関数としてモデル化できる。

(2)ＳＰＮモデルの作成

磁気テープライブラリ装置の制御方式を評価するためのＳＰＮモデルを図 3.4 に示す。図 3.4 において、破線で囲んだ部分がアクセッサによるジョブの並列処理とそれに伴う干渉の発生を表す。プレースp_0はドライブ装置およびアクセッサの実行待ち状態であり、p_1はアクセッサで運搬中の状態、p_2はドライブ装置で処理中の状態である。また、プレースp_3、p_4、p_5は、各々、待ち行列の空席の数、空き状態のアクセッサ台数、空き状

態のドライブ装置台数を示す。

一方、トランジション t_0 はサブシステムへのジョブの到着を表し、t_1 はドライブ装置とアクセッサが両方空いている時にアクセッサにジョブを割り当てることを表す。t_2 はアクセッサの運搬ジョブ終了を表すと同時にドライブ装置の入出力処理開始を表す。t_3 はドライブ装置の入出力処理終了を表す。これらのうち、t_1 は条件が満たされれば瞬間的に発火するが、t_0、t_2、t_3 は遅延を伴う。その遅延時間はランダムであり、指数分布で近似できるものとする。そして、λ_0 は処理要求の平均到着率、$m_1 \cdot \lambda_1$ はアクセッサ群全体の処理性能、$m_2 \cdot \lambda_2$ はドライブ装置群全体の処理性能を示す。λ_1 はアクセッサ1台分の処理性能、すなわち、(1)で求めた性能関数$(TP)^{-1}$であり、m_1 はアクセッサの稼働台数である。また、λ_2 はドライブ装置1個分の処理性能であり、m_2 はドライブ装置の稼働台数である。

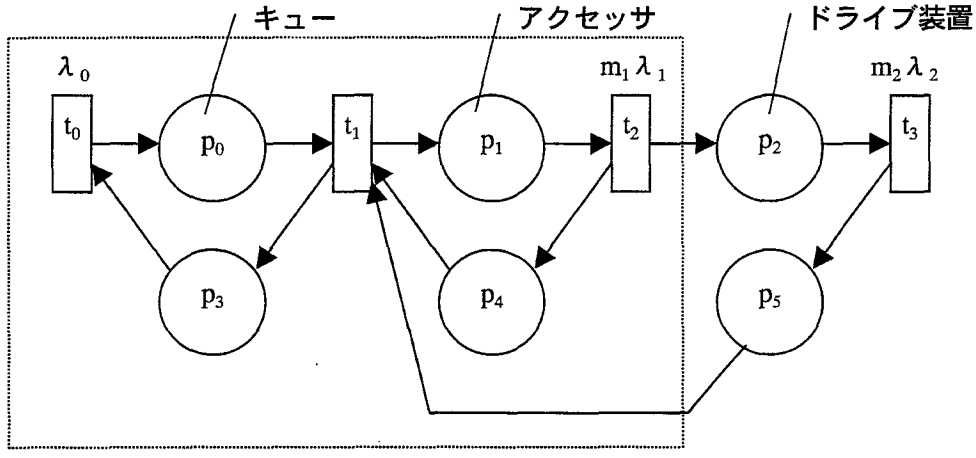


図 3.4　磁気テープライブラリ装置の SPN モデル

(3) モデルの解析

a. 可到達グラフの作成：図 3.4 のモデルを次々に発火させて、可到達グラフを作成する。その結果を図 3.5 に示す。プレース p_3, p_4, p_5 のトークン数は、各々、p_0, p_1, p_2 のトークン数に連動して決まるため、図 3.5 では p_0, p_1, p_2 のトークン数だけを示している。作成した可到達グラフはマルコフモデルにおける状態遷移図を示す。

b. 存在確率の算出：図 3.5 の状態遷移図では、ある状態から他の状態への遷移速度の総和は、他の状態からその状態への遷移速度の総和に等しい、という保存則が成り立つ[50]。この関係に基づいて多元一次連立方程式をたてて各状態の存在確率を求める。

c. スループットおよびターンアラウンドの算出：前記 b で求めた状態の存在確率をもと

に、着目している系内、すなわち、待ち行列プレースp_0とアクセッサプレースp_1に存在する平均トークン数Nを求める。次に、着目している系が単位時間あたり何個の要求を処理できるか、すなわち、系のスループットλを求める。この時、バッファプレースp_0内のトークン数がその容量に達している場合には、系外から要求が到着しても受け付けられないことに注意する。最後に、Little の公式を適用して、次のようにターンアラウンドを算出する。

$$T = N / \lambda \tag{3.7}$$

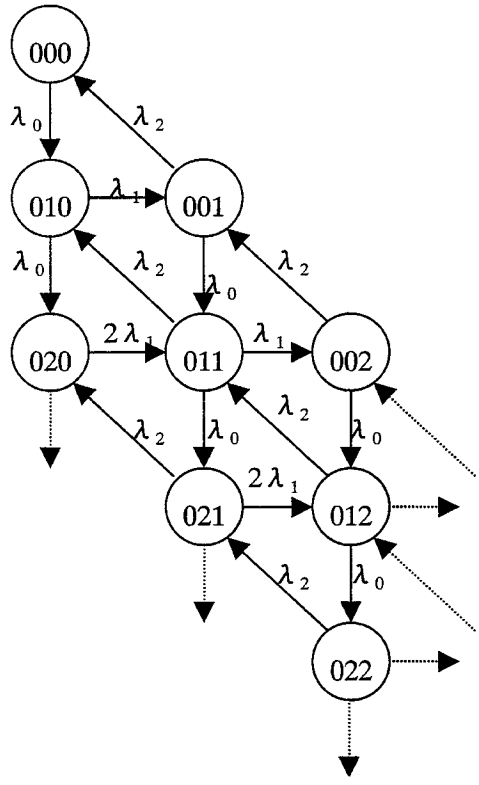


図 3.5　SPN モデルの状態遷移図

3.4.2　提案解析方式の効果

まず、提案した解析方式による評価結果の妥当性について述べ、次に、モデル縮退の効果を述べる。

(1) 評価結果の妥当性

既に述べたように、磁気テープライブラリの制御方式は、①ドライブ装置の最適選択、②アクセッサへの最適ジョブ割当、の２つのステップにより構成される。制御方式の評価を行うためには、これら２つのステップの効果を各々評価することが必要である。そこで、表3.1に示す３つの制御方式についてシミュレーション結果と提案解析方式の比較を行う。

表 3.1　妥当性評価の方法

#	比較対象とする制御方式	ドライブ装置の選択	アクセッサへのジョブ割当
1	最適化方式	最適選択	最適ジョブ割当
2	準最適化方式	ランダムに選択	最適ジョブ割当
3	従来方式	ランダムに選択	先入れ先出し

図 3.6 にスループットの評価結果を示す。本グラフにおいて、横軸は入出力要求の到着頻度を示し、縦軸はこの要求をどれだけ処理できるか、すなわち、スループットを示す。これによれば、最適化方式と準最適化方式の場合、提案解析方式の結果とシミュレーション結果はほぼ等しい。一方、従来方式の場合、提案解析方式の評価結果はシミュレーション結果に比べて若干よいものとなっている。これは、提案解析方式では、運搬ジョブの干渉に伴うアクセッサの加速、減速のオーバヘッドを考慮していないためである。今回の場合、このオーバヘッドは無視しうるが、従来方式と他の方式との微妙な差異を評価したい場合にはオーバヘッドを性能関数に組込む必要がある。以上により、提案解析方式の評価結果はシミュレーション結果とほぼ一致しており、妥当性があることがわかる。

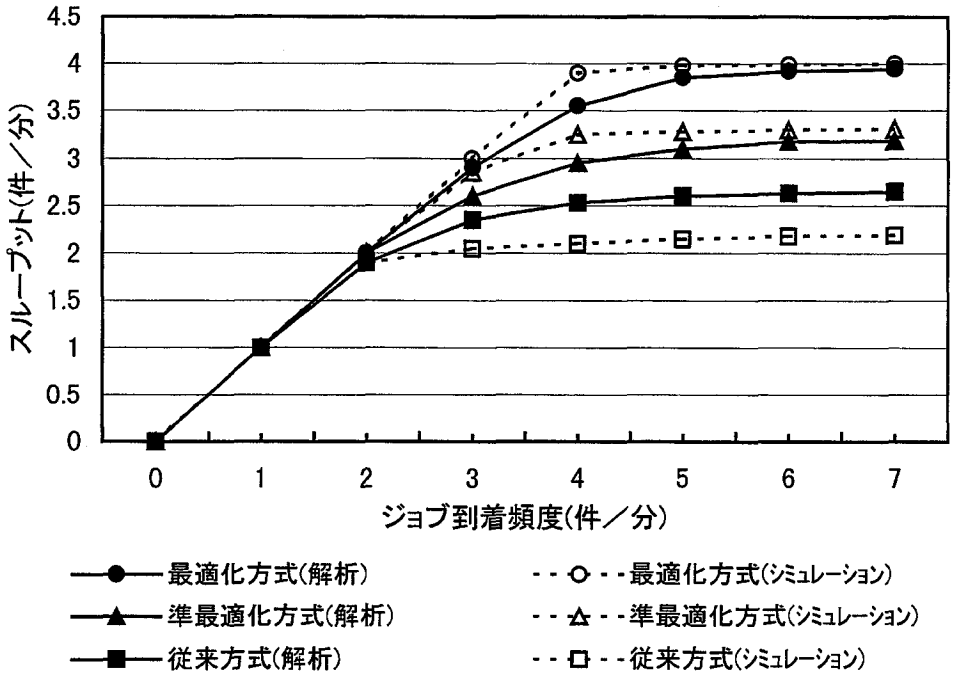


図 3.6　スループットの評価結果

(2) モデル縮退の効果

提案解析方式を適用した場合に、状態遷移空間をどれだけ縮退できるかを評価する。提案解析方式を使わない場合、図 3.4 に示したＳＰＮモデルの点線で囲んだ部分は、図 3.7 に示す SPN モデルに書き換える必要がある。この図において、プレース p_{01}、p_{02}、p_{03} はキューで待ち状態のジョブを示し、各々、3.4.1 項におけるＬグループ、Ｗグループ、Ｒグループに属するジョブを表わす。プレース p_{11}、p_{12}、p_{13} は右アクセッサで運搬中の状態、p_{14}、p_{15}、p_{16} は左アクセッサで運搬中の状態を示す。また、プレース p_q はジョブキュー容量の上限、p_{m1}、p_{m2} はアクセッサ１、２の使用権、p_{r1}、p_{r2} は右テリトリ、左テリトリの占有権を示す。トランジション t_{11}、t_{12}、t_{13}、t_{14}、t_{15}、t_{16} の発火条件は、アクセッサに対するジョブのスケジューリング論理を示す。

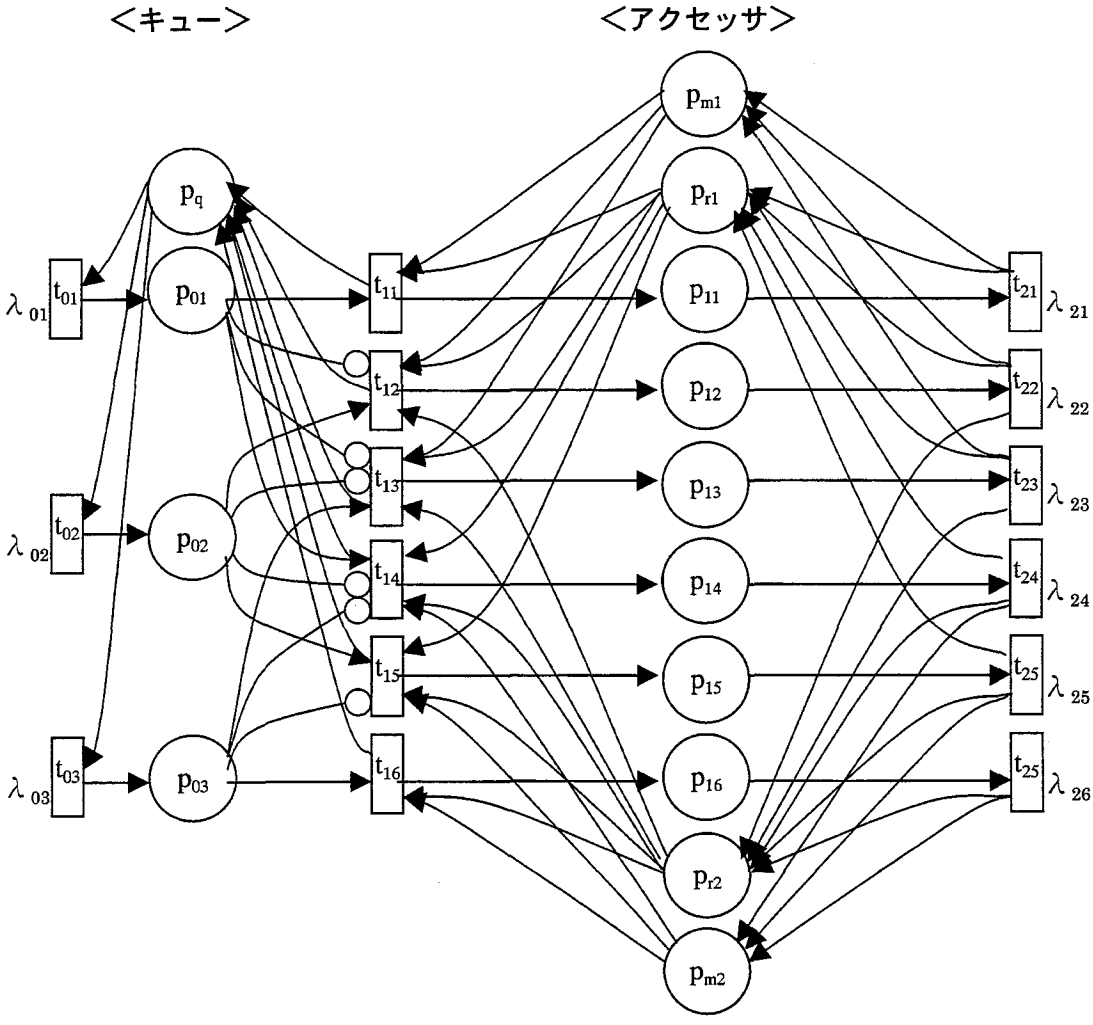


図 3.7　提案方式を使わない場合の SPN モデル

図 3.7 に示すように、提案解析方式を使用しない場合、図 3.4 に比べて膨大な量のプレースとトランジションを必要とする。その結果、提案解析方式によるモデル（図 3.4）の状態数は 228 であるのに対して、図 3.7 のモデルの状態数は約 95,000 となる。すなわち、提案解析方式によれば状態数を約 1/400 に削減でき、実用的な工数でスケジューリング方式の評価を行うことが可能となる。これは、図 3.7 ではスケジューリング論理を厳密にモデル化しているに対して、提案解析方式では性能関数を導入してスケジューリング方式の性能面の相違だけを考慮するからである。

以上に述べたように、提案解析方式によれば、スケジューリング方式を性能関数でモデル化し、マルコフ過程モデルに帰着して解析的に性能評価を行うことができる。従って、評価工数はシミュレーションに比べて極めて低くなる。また、確率ペトリネットを適用して対象システムのモデル化を行うため、システム状態に応じたジョブの条件分岐、同期、処理時間の決定等、システムの動的な特性を表現できる。このため、待ち行列ネットワークに比べて、スケジューリング論理の差異を詳細に解析することができる。

本章で提案した性能関数方式を適用する上で最も重要なポイントは、評価対象とするスケジューリング方式にうまく適合する性能関数の設計と、その妥当性の評価である。一般に、磁気テープライブラリ装置のようなマイクロプロセッサを応用した制御システムでは、リアルタイム性の制約によりディスパッチング形式の単純なスケジューリング方式を採用することが多い。このため、その性能関数は比較的簡単な確率計算により設計可能である。また、同様の理由により、性能関数自体の妥当性は机上で検証できる。従って、提案方式はこのような装置制御システムのスケジューリング方式の解析に適する。ただし、性能関数を組込んだシステム全体の挙動の妥当性検証は、例えば、シミュレーションのような他の方式との比較に頼らざるを得ない。従って、スケジューリング方式の概略設計には簡易な性能関数方式を適用し、詳細設計には正確なシミュレーションを適用するという使い分けが必要であろう。

3.5 結言

本章では、典型的な分散処理システムにおいて課題となるジョブの干渉回避問題を対象として、ヒューリスティックなスケジューリング方式の解析方法を検討した。提案方式では、スケジューリング方式の代替案を、システム状態と処理性能の関係を示す性能関数としてモデル化した。そして、分散処理システム全体を確率ペトリネットによりモデル化し、性能関数を確率ペトリネットモデルに組み込んだ。この方法により、ジョブ管理方式の詳

細なアルゴリズムをモデル化する必要がなくなったため、解空間の規模を大幅に縮小することができ、シミュレーションよりも少ない工数でヒューリスティックス方式の評価を行うことができた。最後に、第２章で論じた磁気テープライブラリ装置のジョブ管理方式の評価を事例として、本解析方式の有効性を示した。

今後の課題は、提案解析方式をさらに複雑なスケジューリング方式の解析に適用することである。例えば、多品種少量生産を対象とする FMS（Flexible Manufacturing System）のスケジューリング、あるいは、多種多様なジョブとリソースを対象とする分散コンピュータシステムのジョブスケジューリング、等である。そのためには、より複雑な性能関数を作成するシステマティックな技法の開発と、作成した性能関数がスケジューリング方式のモデルとして妥当であるかの検証方法の開発が必要となる。

第4章　高メンテナンス性を指向したワークフローシステム構築技法

4.1　緒言

本章では、ワークフローシステムがビジネス環境の変化に柔軟に追従することを課題として、ジョブ管理システムのメンテナンス性を向上するためのシステム構築技法を議論する[68,69,70,71]。

ビジネス分野でインターネットの利用が急速に発展し、企業内および企業間の個人、組織が、コンピュータネットワークにより有機的に結ばれつつある。このような、ネットワークで繋がれた個人が協調して一貫したビジネスプロセスを形成するためのコンピュータツールとして注目されているのがワークフローである[11,12,13,14,15]。ワークフローは、当初、紙ベースでやりとりされていたドキュメント類をイメージデータ化し、ネットワークを介して自動的に回覧するものであったが、現在、これが急速な進化を遂げ、企業の基幹業務をシステム化するためのツールとなりつつある。本ツールにより、事務処理業務を、要素作業（タスク）の実行手順の設計という観点から工学的に扱うことが可能となった。ワークフローをベースとして、種々のタスクに対応するアプリケーションプログラムを実装したシステムをワークフローシステムとよぶ。本章は、このワークフローシステムの構築技法に関するものである。

ワークフローシステム構築技法は、現在までにいくつか提案されている。これらは、ソフトウェア工学に基づくものと CSCW（Computer Supported Cooperative Work）に基づくものに分類できる。前者のソフトウェア工学アプローチは、ビジネスプロセスを構成するタスクをソフトウェアの処理要素と同等であると見なし、フローチャート、ペトリネット等のソフトウェア設計モデルと技法を活用するものである[32,33,72]。これらのモデルと技法を拡張し、タスクの目的、制約、実行主体等を記述するモデルを付加した IDEF（Integration Definition for Function Modeling）技法が開発された[29]。また、ビジネスに関る物、人、金等のオブジェクトと、ビジネスプロセスとの関係をモデル化するユースケース技法も提案されている[30,31]。ここでは、対象ビジネスプロセスとその外部環境であ

るアクター（顧客、仕入先、パートナー、等）を区別し、ビジネスプロセスとアクターの相互作用のために必要なタスクだけをビジネスプロセスの要素とする。

一方、後者の CSCW アプローチは、人間が主体となるビジネスプロセスにソフトウェア工学を適用する限界を指摘した上で提案された。Stanford 大学の Winograd は、言語／行為パースペクティブ理論に基づき、対象業務の関与者が最大限に協調作業を行なえるようなワークフローシステム構築技法を提案している[73,74,75]。

以上のような状況において最近クローズアップされているのが、ワークフローシステムのメンテナンス性の問題である。すなわち、激しいビジネス環境の変化にワークフローシステムをいかに追従させるのか、という改革容易性である。マーケットニーズの変化、新技術の開発等により、ビジネスプロセスが達成すべき目標は変化する。通常、これに対処するために、規則、組織、人的資源という人間系に関わる制約を調整しつつ、タスクの割当や実行順序を変更する。ワークフローシステムの特徴は、ビジネスプロセスを構成するタスク群の作業者割当や実行順序に関する論理を、個々のタスクの処理仕様と分離して定義できることである。このため、ワークフローシステムは、従来のアプリケーションシステムに比べて、高いメンテナンス性を期待できる。ただし、個々のタスクの処理仕様は、他のタスクの処理結果を前提とすることが多いため、実行順序と全く独立に定義することはできない。従って、ワークフローシステム全体のメンテナンス性を確保するためには、タスクの順序制約のうち本質的で変化し難い安定構造を明らかにし、それに基づいてタスクの処理仕様を決定することが課題となる。この問題の解決は、現状では経験的な方法に頼っている。

本章では、ビジネスプロセスの安定構造の抽出と、それに基づくシステムのメンテナンス性の確保を課題とする。そして、それを解決するためのワークフローシステム構築技法を提案する。

4.2 ワークフローシステム構築の課題

まず、一般的なワークフローシステム構築手順を説明し、その中心となるビジネスプロセス設計問題を定義する。そして、ワークフローシステムのメンテナンス性を向上するための課題を示す。

4.2.1 ワークフローシステム構築手順

ワークフローは、ビジネスプロセスを自動化するためのソフトウェアツールである[13]。

ここで、ビジネスプロセスとは、販売、研究開発、資材調達、等のあるまとまった業務を達成するための要素作業の繋がりである。従来、企業のオフィス業務は調査、検討、承認、等の人間系作業が主体となっており、計算機による自動化の対象となり難かった。会計処理、顧客管理処理、等のような一部の定型的な業務機能がシステム化されたが、それらの機能は互いに分断されており全体業務としての効率化に繋がらなかった。ワークフローを適用することにより、それらの既に自動化された業務機能と未だ自動化されていない人間系作業を統合して、仕事の流れ全体を自動化し管理することができる。

ワークフローツールを中心としたアプリケーションシステムをワークフローシステムとよび、その全体イメージを図 4.1 に従って説明する。

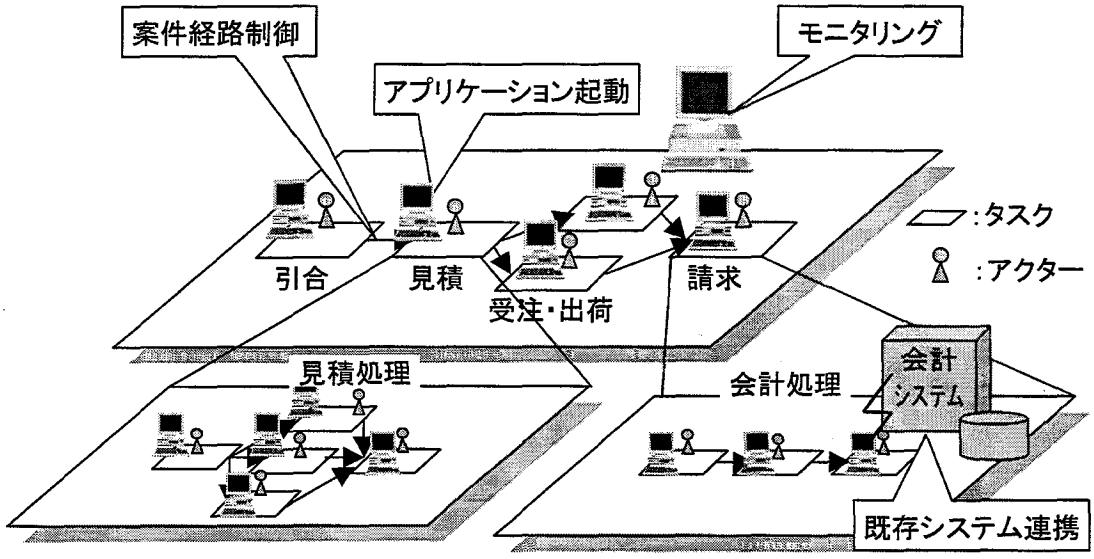


図 4.1　ワークフローシステムのイメージ

既に述べたように、ワークフローシステム構築の目的は、対象とするビジネスプロセス全体の実行管理である。従って、システムの主たる機能は、ビジネスプロセスで実行される業務案件すなわちジョブの経路制御と進捗モニタリングである。また、必要に応じて、個々のタスクを支援するためのアプリケーションを起動し、既存システムとの連携を行う。このため、アプリケーション処理は二義的なものであり、通常、人間系に作業の到着をしらせるための画面処理と、既存システムに作業結果を受け渡すためのデータベース入出力処理に留まる[16]。ワークフローシステム構築において中心となる行為は、現行業務を分析した上で、与えられた目標と現実とのギャップを埋めるために新たなビジネスプロセスを設計することである。

このような理由から、ワークフローシステムの構築では、従来のアプリケーションシステム構築における機能とデータという設計観点[32]に加えて、ビジネスプロセスという新たな観点が重要となる。ワークフローシステム構築での機能とは、個々のタスクを支援するアプリケーションの処理仕様を示す。既に述べたように、画面処理とデータベース入出力処理のアプリケーションが主体となる。データとは、上記のアプリケーションを介して既存システムと連携を行うためのデータベース仕様を示す。既存システムからの情報参照、あるいは、既存システムへの作業結果の受渡しを目的とした二次ファイルが主体となる。プロセスとは、対象業務を複数のタスクに分割し、それらの順序関係を設定し、実行担当者を割り当てることを示す。この設計結果はワークフローツールで実装されるとともに、アプリケーションの処理仕様とデータベース仕様を設計するための基本情報となる。一方、システム構築の手順は、業務設計、システム設計、実装という標準的な３つのフェーズから成り、図 4.2 に示すように、各構築フェーズごとにプロセス、機能、データの３つの観点による設計を行う。以下、ワークフローシステム構築手順をこの図に従って説明する。

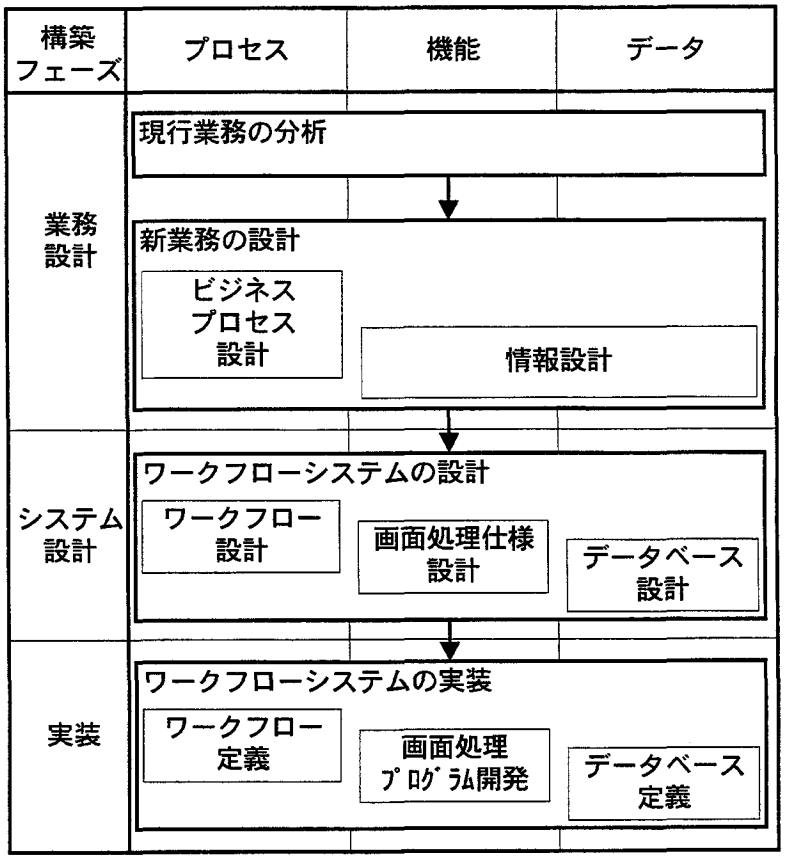


図 4.2　ワークフローシステム構築手順

(1) **業務設計フェーズ**

本フェーズでは、現行業務の分析と、その結果に基づく新業務の設計を行う。

①現行業務の分析：対象業務の現状調査を行い、その業務の遂行に必要な作業（タスク）群とそれらの実行手順（オーダー）、および、実行担当者（アクター）を明らかにする。さらに、社内規則、組織の権限と義務、利用可能な資源、等の制約条件を明らかにする。

②新業務の設計：プロセスの設計観点からビジネスプロセス設計を、機能とデータの設計観点から情報設計を行う。ビジネスプロセス設計では、対象業務が与えられた制約条件のもとで目標を実現できるように、タスク、オーダー、アクターの関係を決定する。これをビジネスプロセス仕様とよぶ。このビジネスプロセス設計は、次項で詳しく説明する。また、情報設計では、ビジネスプロセス設計の結果明らかにしたタスク群の入出力情報を整理し、その情報を構成する実体群とその関連を設計する。これを情報仕様とよぶ。

(2) **システム設計フェーズ**

プロセス、機能、データの設計観点に基づいて、各々、ワークフロー設計、画面処理仕様設計、データベース設計を行う。

①ワークフロー設計：ビジネスプロセス仕様に基づき、ワークフローツールで定義すべき次の情報を設計する。

・ワークフロー定義情報：タスクとオーダーを、ワークフローモデルにおけるノードとその接続関係としてモデル化する。また、アクターを、各ノードの実行担当者の役割名（ロール）としてモデル化する。さらに、開始、終了、作業、分岐、等の各ノードに対して、名称、起動アプリケーション、判定基準、等の属性値を設計する。

・ユーザ／ロール情報：ワークフローシステムの全ユーザに対して、対象業務を遂行するための役割名であるロールを割り当てる。

②画面処理仕様設計：情報仕様に基づき、ワークフローの各ノードで起動する画面処理プログラムの処理仕様を設計する。

③データベース設計：上記の画面処理プログラムで利用するデータを格納するために、そのデータベースの論理設計を行う。

(3) **実装フェーズ**

ワークフロー定義、画面処理プログラム開発、データベース定義を行う。ワークフロー定義は、上記のワークフロー設計結果を市販のワークフロー定義ツールで設定する。デー

タベース定義は、上記のデータベース論理設計の結果を市販のデータベース定義ツールで設定する。画面処理プログラム開発は、市販の帳票作成ツールまたはプログラミング言語を利用して行う。

4.2.2 ビジネスプロセス設計問題

前項で述べたように、ワークフローシステムの構築では、業務設計フェーズのビジネスプロセス設計が最も重要な行為であり、その後のフェーズでワークフロー、アプリケーション、データベースの仕様を決定するものである。ここでは、ビジネスプロセス設計を更に詳細に検討し、その技術課題を明らかにする。

図 4.3 に示すように、ビジネスプロセス設計とは、対象業務が与えられた制約条件のもとで目標を実現できるように、タスク、オーダー、アクターの関係、すなわち、ビジネスプロセス仕様を決定することである。ここで、タスクは対象業務を複数人が分担して実行するための機能単位であり、その実行主体であるアクターの組織規模、実行能力により割り当てるべき機能階層が異なる。すなわち、組織規模が大きいほど、あるいは、実行能力が高いほど、高度な機能のタスクを割り当てる。

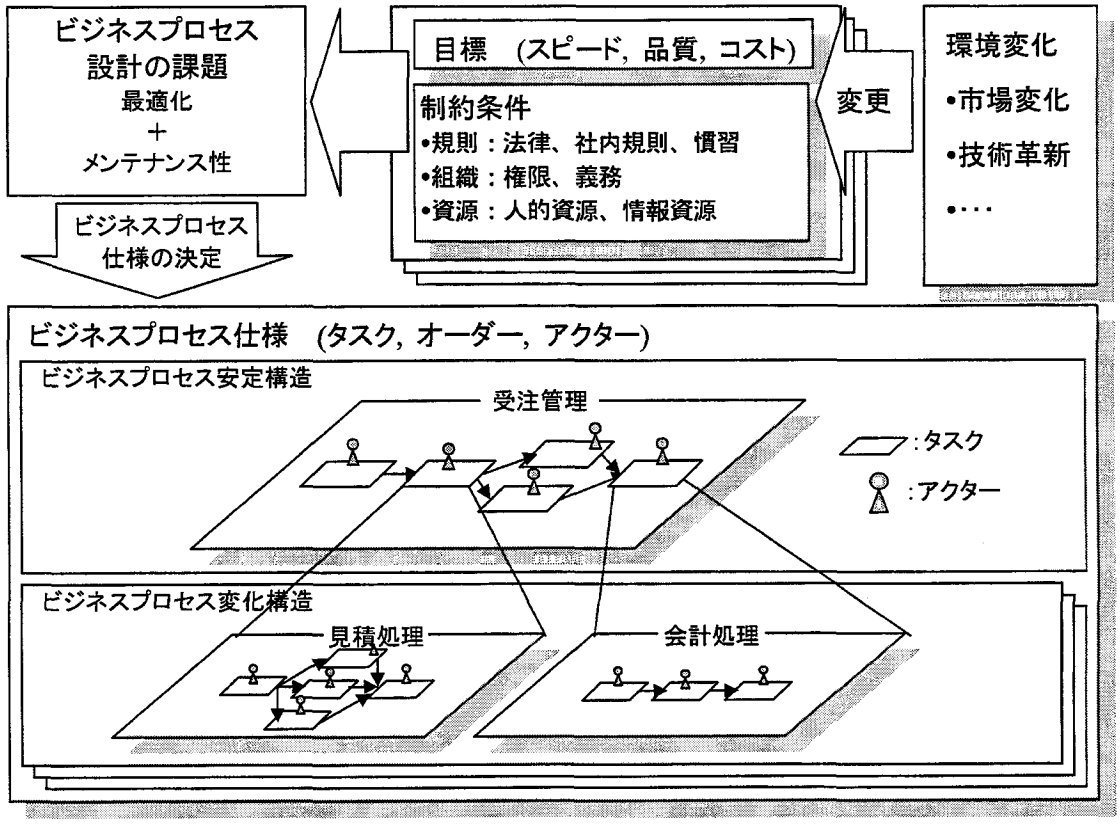


図 4.3 ビジネスプロセス設計問題

ビジネスプロセス設計における目標は、顧客に対して製品またはサービスを提供する際のスピード、品質、コストに関するものである。一方、制約条件は、規則、組織、資源の３種類に分類できる。規則とは、法律、社内規則、慣習、等を示す。規則の強制力には段階があり、公的なものほど厳しく私的なものほど緩和し易い。組織制約とは、組織の有する権限と責任を示す。業務の担当者はこの組織制約の範囲で選ぶ必要がある。資源制約とは、業務遂行のために供給可能な人的資源と情報資源を示す。人的資源は情報資源により部分的に代替可能である。一般に、ビジネスプロセス設計問題では、制約条件が厳しすぎるため、制約条件を満足する解は存在しないことが多い。従って、与えられた目標を満足するビジネスプロセスを設計するためには、制約条件を緩和することが必要となる。

ここで注意することは、今日のように市場、技術等に関する環境変化が激しい状況では、目標と制約条件が頻繁に変更されることである。このため、ある時点で最適にビジネスプロセスを設計してワークフローシステムを構築しても、その後の環境変化に伴い、様々な不具合が生じることが多い。ビジネスプロセス設計では、このような環境変化を見通してその対策を考えることも必要となる。従って、ビジネスプロセス設計問題は、与えられた目標と制約条件を満足するようにビジネスプロセス仕様を決定する最適化問題であると同時に、環境変化に柔軟に対応するためのメンテナンス性向上の問題でもある。

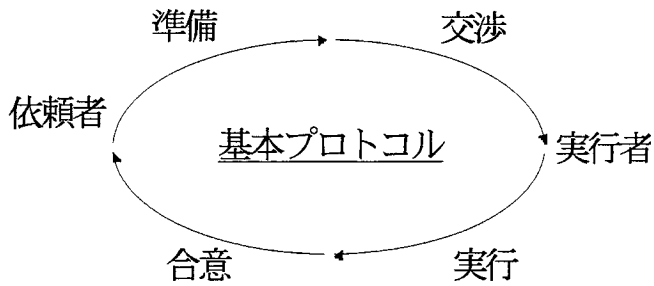
4.2.3 メンテナンス性向上の課題

ワークフローシステムのメンテナンス性を向上するためには、ビジネスプロセス仕様の中から環境変化の影響を受け難い安定部分（ビジネスプロセス安定構造）を抽出し、その他の変化し易い部分（ビジネスプロセス変化構造）と明確に分離して、ワークフローシステムの設計と実装を行えばよい。

本章では、ビジネスプロセス安定構造を見出すために、タスクの機能階層に着目する。前項において、タスクは対象業務を複数人が分担して実行するための機能単位であり、その実行主体であるアクターの組織規模、実行能力により割り当てるべき機能階層が異なることを述べた。さらに、ビジネスプロセス仕様を決定する重要な要素である制約条件は、組織レベルのものほど緩和し難く、個人レベルのものほど変わり易いことを述べた。従って、タスクの機能階層を高いレベルにとれば、そのビジネスプロセス構造は極めて安定したものとなる。一方、ワークフローシステム構築のためには、対象業務を複数の実行主体が分担して実行できるレベルにタスクを分割する必要がある。従って、ビジネスプロセス安定構造を抽出するためには、タスク機能階層を最適に設定することが必要となる。

4.2.4 タスク階層モデル

ワークフローシステムのメンテナンス性を確保する上で中心的な課題となるタスク階層の問題を解決するために、 Winograd のタスク階層モデルを検討する。このモデルは、ビジネスプロセスにおける依頼者と実行者の委託関係に注目し、タスクとアクターの階層化を行なったものである。以下、このモデルの概略を説明する。



#	状態	行為の内容
1	準備	依頼者が実行者に対して委託すべき業務の仕様を提示する。
2	交渉	業務仕様に関して、依頼者と実行者との間で合意をとる。合意がとれない場合、仕様の再提示を行う。
3	実行	実行者は、合意された仕様に従って業務を実施し、その成果物を依頼者に渡す。
4	合意	依頼者は、業務仕様に従って成果物を評価する。満足できない場合、実行者に再実行を依頼する。

図 4.4 業務委託の基本プロトコル

ビジネスプロセスには、業務の依頼者と実行者の二者が常に存在する。依頼者は実行者に対してある条件のもとに業務を委託する。それをトリガーとして、図 4.4 に示す、準備、交渉、実行、合意という状態遷移に従ってプロセスが進行する。そして、各状態において、依頼者と実行者が、業務の依頼、実行条件の調整、実行結果の通知等の行為を行う。状態遷移は、これらの行為の実行結果に応じて決まる。この状態遷移モデルを、ビジネスプロセスの基本プロトコルとよぶ。基本プロトコルは、業務内容にかかわらず一定である。

実際の業務は、依頼者と実行者の二者だけで実行できるほど単純で小規模なものだけではない。そこで、前記の基本プロトコルに従って分割したタスクを他の担当者に委託する。その結果、ビジネスプロセスは、基本プロトコルが再帰的に組み合わさった図 4.5 のよう

な委託関係のネットワークとなる。すなわち、対象業務を達成するために設定した下位レベルのタスクも、基本プロトコルと同じ状態遷移に従ってプロセスが進行する。図において、真中にある最も大きな楕円が、対象とする業務の最上位のプロトコルを示す。その依頼者と実行者は、業務の実行結果に最も責任をもつ人とする。その周りに矢印で繋がれた小さな楕円は、各状態において第三者に依頼するタスクを示す。これらのタスクもまた依頼者と実行者を有し、基本プロトコルに従って実行される。以降、レベルが下位になるに従って楕円の大きさが小さくなる。異なるレベルの楕円が矢印で繋がっている場合、大きな楕円で行うべきタスクを小さな楕円に分割し委託していることを示す。タスクの実行順序は、準備状態に属するタスクが最も早く実行され、その後矢印に沿って時計回りに順次実行される。

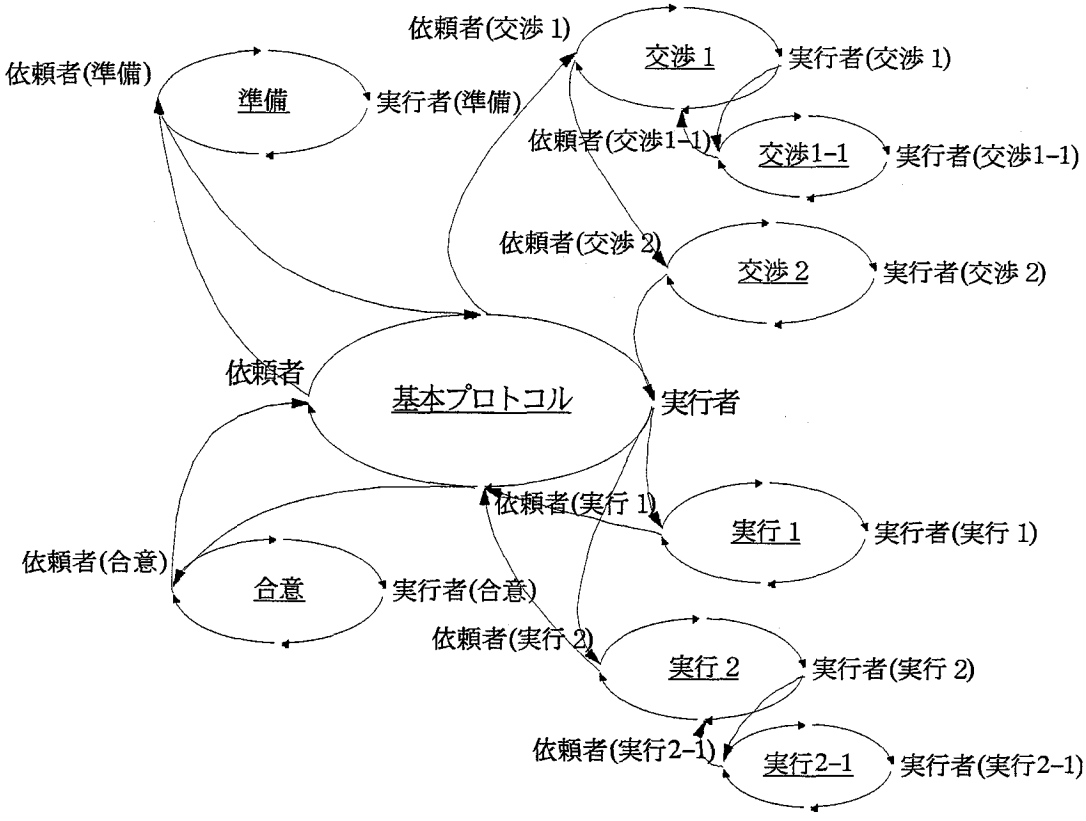


図 4.5　委託関係ネットワーク

このように、Winograd のモデルでは、タスクの委託関係に基づいてその階層を設定する。このタスク階層モデルを適用することにより、対象とするビジネスプロセスの安定構造を導き、それに基づいてワークフローシステムを構築する方法を提案する。

4.3 メンテナンス指向ワークフローシステム構築技法

4.3.1 基本方針

ビジネスプロセスは、業務の依頼者が実行者に対して要求を出し、その要求を実現するために多くの人によって種々の作業が実行され、最終的に依頼者がその実行結果を受理するまでの一連の過程である[8]。前節で述べたように、ビジネスプロセスには、対象業務の実行に最も責任のある依頼者と実行者が存在する。これらをオリジナルの実行者と依頼者とよぶ。オリジナルの依頼者と実行者は、業務をいくつかのタスクに分割して協同で実行する。この時、独力で実行できないタスクがある場合、それを第三者に委託する。このような委託関係を繰返すことにより、ビジネスプロセスは複雑で巨大なものとなる。

従って、「業務がオリジナルの依頼者と実行者だけによって実行される」と仮定することにより、本質的なビジネスプロセス安定構造を得ることができる。なぜなら、業務をこれら二者間の相互作用（インタラクション）とみることにより、頼む／頼まれるという委託関係が必要最小限のものとなり，その結果、規則、組織、資源という制約条件の影響を最小化できるからである。すなわち，ビジネスプロセスの安定構造は、オリジナルの依頼者と実行者の二者間で実行するタスクとオーダーである。

そこで、二者間のインタラクションを仮定して生成するビジネスプロセスをプロセスパターン、それを構成するタスクを基本タスクと呼ぶことにする。このプロセスパターンを、対象とするビジネスプロセスを設計する際のフレームワークととらえ、システム設計フェーズでワークフロー設計を行う時には、それ以外の環境変化の影響を受け易い部分と明確に分離する。

次に、プロセスパターンを適用して、メンテナンス性の高いアプリケーションとデータベースを設計する方法を述べる。一般に、アプリケーションとデータベースのメンテナンス性は、それらを安定したデータ構造に基づいて開発することにより向上できる。プロセスパターンとして抽出されたタスクとそのオーダーは安定したものであるから、各タスクの入出力情報（帳票、ファイル、等）は、対象業務にとって必要かつ十分な情報を示す。そして、この情報を構成する実体群の関連を求めることにより、情報に関する安定構造を得ることができる。このようなプロセスパターンに基づく情報安定構造を情報パターンとよぶことにする。この情報パターンに基づいてアプリケーションとデータベースを設計することにより、環境変化の影響を受け難い安定仕様と、そうでない変化仕様を分離することができる。

以上に述べたプロセスパターンと情報パターンを総称して設計パターンとよぶことにす

る。以下、設計パターンの抽出方法と、それに基づくワークフローシステム構築手順を説明する。

4.3.2 プロセスパターンの抽出方法

プロセスパターンを導き出すために、現行プロセスのタスクの委託関係を、図 4.4 および図 4.5 に示す Winograd のモデルを用いることにより明らかにする。まず、オリジナルの依頼者と実行者を見出す。そして、基本プロトコルの状態遷移（準備、交渉、実行、合意）に従って、業務を達成するために必要なタスクと、それを委託する相手を明確にする。その委託相手がさらにタスクを分割して他の人に委託するならば、それを明らかにする。

次に、その結果に二者間のインタラクションを仮定して、対象業務をオリジナルの依頼者と実行者の二人だけで実施する場合のタスク、アクター、オーダーを、次のように決定する（表 4.1 を参照）。

表 4.1 プロセスパターンの抽出方法

#	決定項目	プロセスパターン抽出方法	
		対象ビジネスプロセスの構成要素	プロセスパターンの構成要素
1	タスク	オリジナルの依頼者または実行者が実行するタスク	プロセスパターンを構成する基本タスクとする。
		オリジナルの依頼者または実行者が第３者に委託するタスク	
		上記以外のタスク	削除する。
2	アクター	オリジナルの依頼者または実行者	同左。
		オリジナルの依頼者または実行者が委託した第３者	委託元である、オリジナルの依頼者または実行者とする。
		上記以外のアクター	削除する。
3	オーダー	オリジナルの依頼者または実行者が実行するタスク、あるいは、彼らが第３者に委託するタスクの実行順序	タスク間に因果関係がある場合には逐次的に繋ぎ、そうでない場合は並列化と統合を検討する。
		上記以外のタスクの実行順序	削除する。

(i) タスク：オリジナルの依頼者と実行者自身が行うタスク（第１レベルのタスク）、および、オリジナルの依頼者と実行者が第三者に直接委託するタスク（第２レベルのタスク）を、プロセスパターンを構成する基本タスクとする。すなわち、図 4.5 における最上位レベルの楕円と第二レベルの楕円で示されるタスクを残し、その他は削除する。

(ii) アクター：上記の方法で選択したタスクは、オリジナルの依頼者または実行者のいず

れかが実行するものとする。すなわち、第１レベルのタスクは、プロセスパターンにおいてもオリジナルの依頼者と実行者がアクターとなる。また、第２レベルのタスクも、第三者ではなく、オリジナルの依頼者または実行者自身がアクターとなる。

(iii) オーダー：基本プロトコルにおいて、準備、交渉、実行、合意の４つの状態間には明確な因果関係がある。従って、各状態に１つずつタスクがある場合には、そのオーダーは４つのタスクが逐次的に繋がったものとなる。一方、１つの状態に複数個のタスクがある場合には、それらのタスク間に因果関係がないことが多い。従って、タスクの並列化と統合を検討する。

図 4.6 は、あるサービス会社の案件の受注から請求、入金までのプロセスモデルを示す。顧客、窓口、営業部門、サービス部門の４部署が関与し、14 人の人間が作業を行う。本プロセスの最も上位レベルの委託関係は真中の最も大きな楕円で示す「サービス案件の実施」であり、依頼者はクライアント、実行者はサービス会社である。

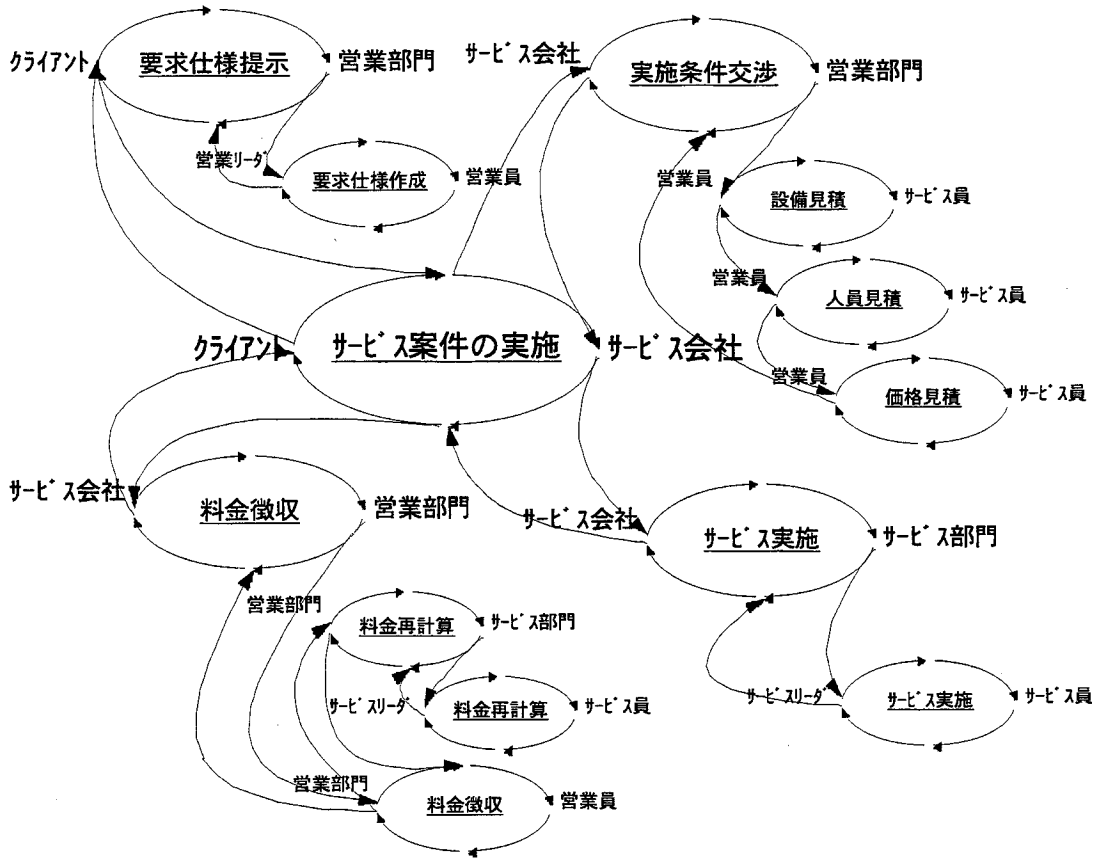


図 4.6　現行業務プロセスモデル（受注管理業務）

図 4.7 は、図 4.6 のプロセスに二者間のインタラクションを仮定して得られたプロセスパターンを示す。基本タスクは、要求仕様提出、実施条件交渉、サービス実施、料金徴収の４つである。アクターはクライアントとサービス会社の２者であるため、４つの基本タスクは依頼者自身が実行する。オーダは上記の４つのタスクがシーケンシャルに繋がる。図 4.7 で示されるタスクのうち、準備状態に属する要求仕様作成タスク、交渉状態に属する見積りタスク等はパターンからは削除されている。それは、プロセスパターンに示す４つのタスクを更にどのように分割して誰に委託するかが、規則、組織、資源に関する制約により変化するからである。

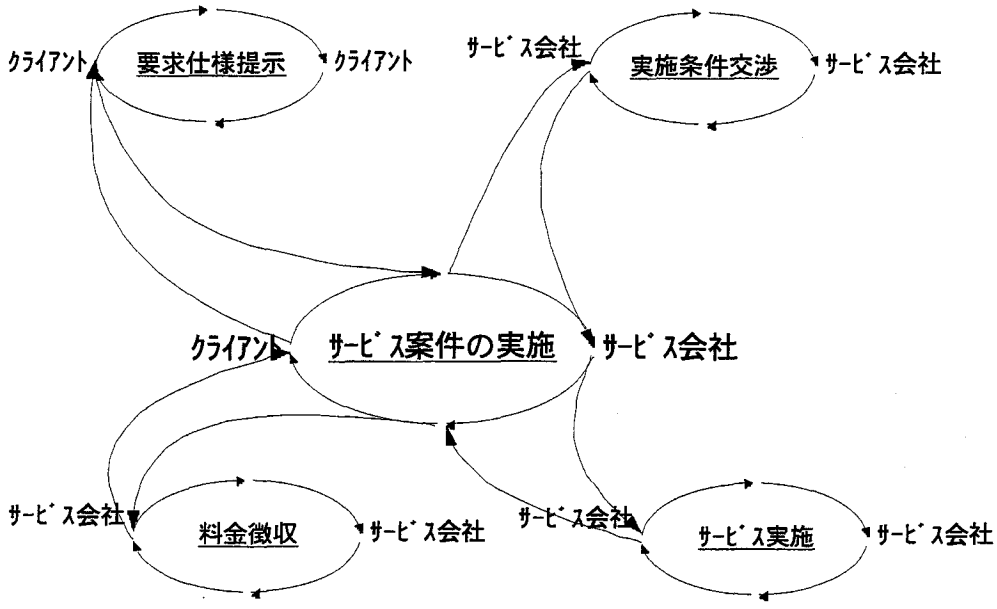


図 4.7　プロセスパターン（受注管理業務）

4.3.3　情報パターンの抽出方法

情報パターンとは、プロセスパターンに属する基本タスクで入出力する情報の実体関連である。基本タスクの入出力情報をもれなく洗い出すためには、ビジネスプロセス上での各タスクの役割を分析することが有効である。タスクの役割は、 Winograd のモデルの準備、交渉、実行、合意の４状態のいずれに属するかによって決まり、表 4.2 に示すように入出力情報の要件を明らかにすることができる。

まず、準備フェーズに属する基本タスクは、対象業務の要求仕様を出力する。次に、交渉フェーズの基本タスクは、この要求仕様を入力して、それを実現するための実行計画を出力する。実行フェーズの基本タスクは、要求仕様と実行計画を入力して、依頼された業務を実施した後に実行報告を出力する。最後に、合意フェーズの基本タスクは、要求仕様

と実行報告を比較して、結果評価を出力する。以上に述べた、要求仕様、実行計画、実行報告、結果評価という４種類の実体は、プロセスパターンにおいて依頼者と実行者の２人の間で授受する必要十分な情報である。

表 4.2　入出力情報の要件

#	項目	役割	入力情報	出力情報
1	準備タスク	依頼者から実行者への依頼内容の提示。	－	要求仕様：依頼する業務の仕様、実施条件等を指定するための情報。
2	交渉タスク	依頼者の合意に基づく実施条件の決定。	要求仕様	実行計画：業務の実施条件を明確にし、依頼者と交渉を行うための情報。
3	実行タスク	実行者による依頼事項の実施。	要求仕様 実行計画	実行報告：依頼される業務の実行完了とその成果を、依頼者に示すための情報。
4	合意タスク	依頼者による実行結果の評価。	要求仕様 実行報告	結果評価：依頼者が、実行者から渡された成果の評価結果を示すための情報。

以上のようにして求められた４種類の実体は、既存のファイル、データベース等に存在する物、金、人に関わる実体（製品、資産、顧客、仕入先等）と密接な関連があることが多い。従って、情報パターンは、プロセスパターンに基づいて求めた４種類の実体と、既存の物、金、人に関わる実体の関連を明らかにすることで抽出することができる。

図 4.8 は、先の受注管理プロセスの情報パターンを示す。入出力情報は、表 4.2 における要求仕様、実行計画、実行報告、結果評価の４種類の実体に対応させることにより、要求仕様、見積書、納入物、実行確認書であると特定できる。クライアントとサービス会社のインタラクションのためには、最低限これらの入出力情報が必要である。さらに、これらの情報を生成するためには、顧客リスト、サービスメニュー、サービス要員ファイル、請求書等の実体が必要となる。本ビジネスプロセスを運用するための帳票、データベース、アプリケーションを開発する際には、この情報パターンをベースとすれば高いメンテナンス性を実現できる。

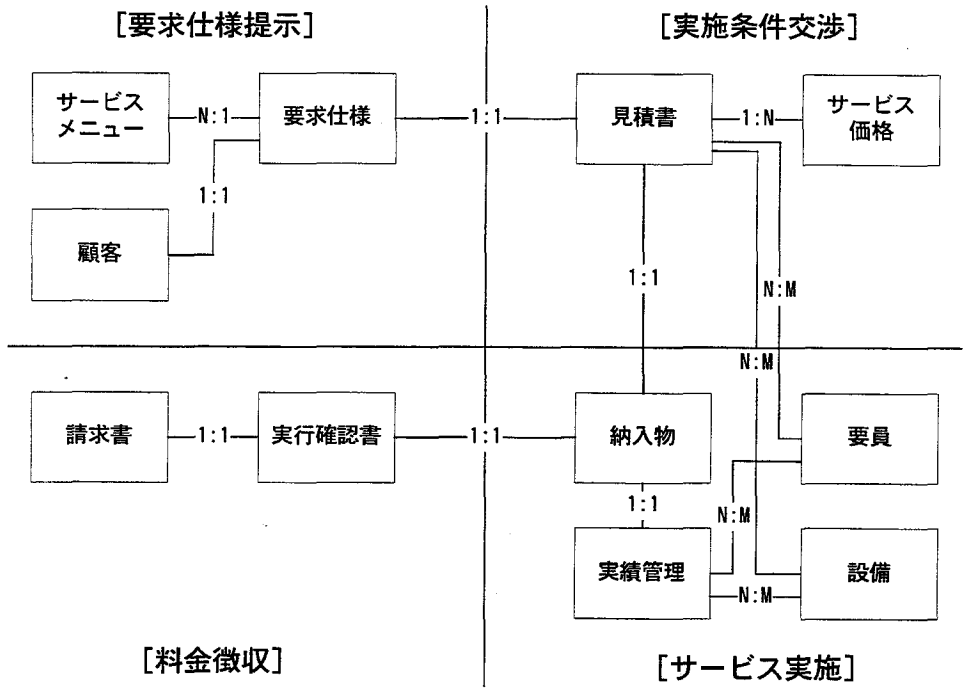


図 4.8　情報パターン（受注管理業務）

4.3.4　設計パターンに基づくワークフローシステム構築手順

プロセスパターンと情報パターンの２つから成る設計パターンをベースに、ワークフローシステムを構築する新たな手順を図 4.9 に従って説明する。図 4.2 で示した一般的な構築手順との違いは、業務設計フェーズにおける現行業務分析と新業務設計の間で設計パターンの抽出を行い、その結果に基づいてシステムの安定仕様と変化仕様を分離して設計することである。

(1) 業務設計フェーズ

現行業務を分析して設計パターンを抽出する方法は既に説明した。ここでは、設計パターンに基づいて、環境変化の影響を受け易い変化部分を設計する方法を示す。

＜プロセス変化構造の設計＞

プロセスパターンとして抽出された基本タスクをさらに詳細なタスクに分割し、各々にアクターを割り当て、オーダーを設定する。具体的には次の方法をとる。

(i) タスク分割とアクターの割当

基本タスクをさらに担当者レベルのサブタスクに分割する。タスク分割の基準は、３つの制約条件である規則、組織、資源に基づく。ビジネスプロセス設計問題の目標を実

現するために、必要に応じて、規則の見直し、組織権限の調整、情報技術の活用、等の策を検討し、タスク分割を必要最小限のものとする。例えば、従来、異なる部署が担当していたタスク群であっても、計算機の支援により統合できる場合、それを検討する。また、書類の集配、登録、データの転送等の定型業務は、計算機により自動化する。

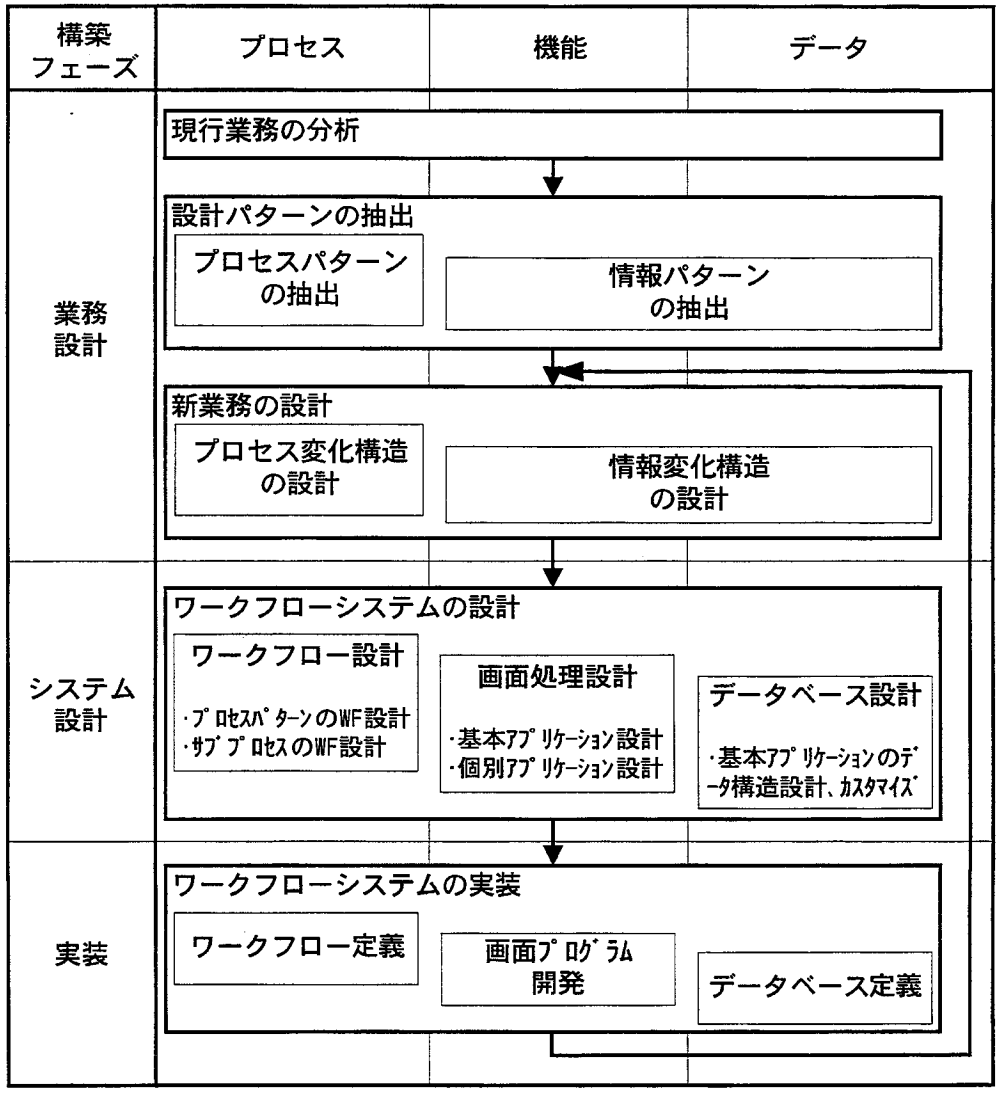


図 4.9　設計パターンに基づくワークフローシステム構築手順

(ii)オーダーの設定

基本タスクのオーダーは既に決まっているため、上記のように設定されたサブタスク間のオーダーだけを新たに決定する。この際、与えられた目標を実現するために、必要に応じて、タスクの並列実行を図ることが必要となる。通常、サブタスク間のオーダー

は3つの制約条件により一意に決まり、自由度がほとんどないことが多い。しかし、これらの制約条件は、従来の紙ベースの事務処理形態を前提として決められており、情報技術を活用したプロセスの場合にはそれらを緩和できることが多い。例えば、申請書類の複数部署によるチェック作業は、ワークフローの分配／同期機能を利用すれば並列化可能である。

＜情報変化構造の設計＞

既に述べたように、情報パターンは基本タスクの実行に必要な情報の構造を定めたものである。プロセス変化構造の設計は基本タスクをサブタスクに分割するものであるため、それに伴い情報パターンを大幅に変更することはない。必要に応じて、サブタスクを委託するための内部管理情報を付加する。ただし、基本タスクの追加／削除、順序関係の変更がある場合は、実体関連の見直しが必要となる。通常、このような情報パターンの大幅な変更は、設計パターンを抽出した直後に実施することは少なく、環境の変化に伴いワークフローシステムをメンテナンスする際に必要となる場合がある。

(2) システム設計フェーズ

＜ワークフロー設計＞

ワークフローツールが標準装備する定義ツールにより以下のワークフロー設計を行う。

(i) プロセスパターンのワークフロー設計

プロセスパターン、すなわち、基本タスクの順序関係を実現するためのワークフロー設計を行う。プロセスパターンは環境変化の影響を受け難い安定仕様をもつ。従って、アプリケーション仕様は、後述するように、プロセスパターンのワークフロー設計結果に基づいて決める。

(ii) サブプロセスのワークフロー設計

基本タスクを他のアクターに委託した結果生じるサブプロセスのワークフロー設計を行う。このサブプロセスはプロセスパターンに比べて環境変化の影響を受け易い。従って、メンテナンス性を向上するために、プロセスパターンとは独立に定義し必要に応じて呼び出される形態にする。

＜画面処理仕様およびデータベースの設計＞

ワークフローにより呼び出される画面処理アプリケーションプログラムの仕様設計を、市販のプログラミング言語、あるいは、帳票作成ツールの適用を想定して、以下のよう

に行う。また、データベース設計は市販のデータベース定義ツールを用いて行う。

(i) 基本アプリケーションの設計

情報パターンに基づいて、基本的タスクを実行するためのアプリケーションプログラムを設計する。このプログラムを基本アプリケーションとよぶことにする。設計対象は、処理、データに関する次の項目である。

・処理：画面レイアウト、入力チェック処理、データ入出力処理、ワークフロー連携処理（次タスクへの遷移、前タスクへの差戻し、保留、等）。

・データ：画面処理データ、定数ファイル等のテーブル構造。

(ii) 個別アプリケーションの設計

新ビジネスプロセスのサブタスクから呼び出す個別アプリケーションの設計を、基本アプリケーションの仕様をベースにそれをカスタマイズすることで行う。主なカスタマイズ項目は以下の通りである。

・情報のアクセス制御：アクターの役割により、参照および更新できる情報を制限する。それに応じて、画面処理プログラムのカスタマイズが発生する。

・情報の付加：タスクを分割し第三者に委託する際に、タスクの内容を示す情報とともに、条件、注意、連絡等のメッセージを付加することが必要となる。また、タスクの実行時に参照する関連情報を添付することも必要となる。

また、必要に応じてデータベース構造のカスタマイズを行う。基本アプリケーションは安定的な処理仕様を有するから、この方法によりアプリケーションプログラムのメンテナンス性が向上する。

(3) 実装フェーズ

上記の設計結果に基づき、ワークフロー定義、画面処理プログラム開発、データベース定義を行う。このうち、画面処理プログラム開発は、市販のプログラミング言語、あるいは、帳票作成ツールの開発環境を利用して行う。また、ワークフロー定義とデータベース定義は、市販の定義ツールを利用することにより、設計結果に少数の実装上の物理パラメータを付加して自動的に生成することができる。

(4) メンテナンス

環境変化に伴いワークフローシステムのメンテナンスが必要になった場合は、設計パターンのカスタマイズで対処することを検討する。これにより、ワークフローの変更はサブプロセスレベルだけでよく、また、アプリケーションの変更は基本アプリケーショ

ンのカスタマイズで済む。従って、メンテナンス工数は極めて小さくなる。

以上に述べたように、設計パターンを活用することにより、ビジネス環境の変化に応じて最適なワークフローシステムを柔軟に構築することができる。

4.4 ケーススタディ

先のサービス会社の受注管理プロセスを例として、本章で提案したワークフローシステム構築技法のメンテナンス性向上に関する効果を示す。ここでは、受注管理プロセスのワークフローシステムを、提案方式により設計した場合と従来通りメンテナンス性を意識せず設計した場合を比較する。

図 4.7、図 4.8 の設計パターンに基づいて設計した新ビジネスプロセスを図 4.10 に示す。本ケーススタディでは、準備、交渉という計画段階の作業はほぼ現行通りのプロセスとし、実行、合意という実現段階の作業は、中間的な指示系統を介さず担当者に直接依頼するように改善する。計画段階はノウハウが必要であり現状では計算機化が困難であるが、実現段階は管理情報の共有により効率化できるためである。

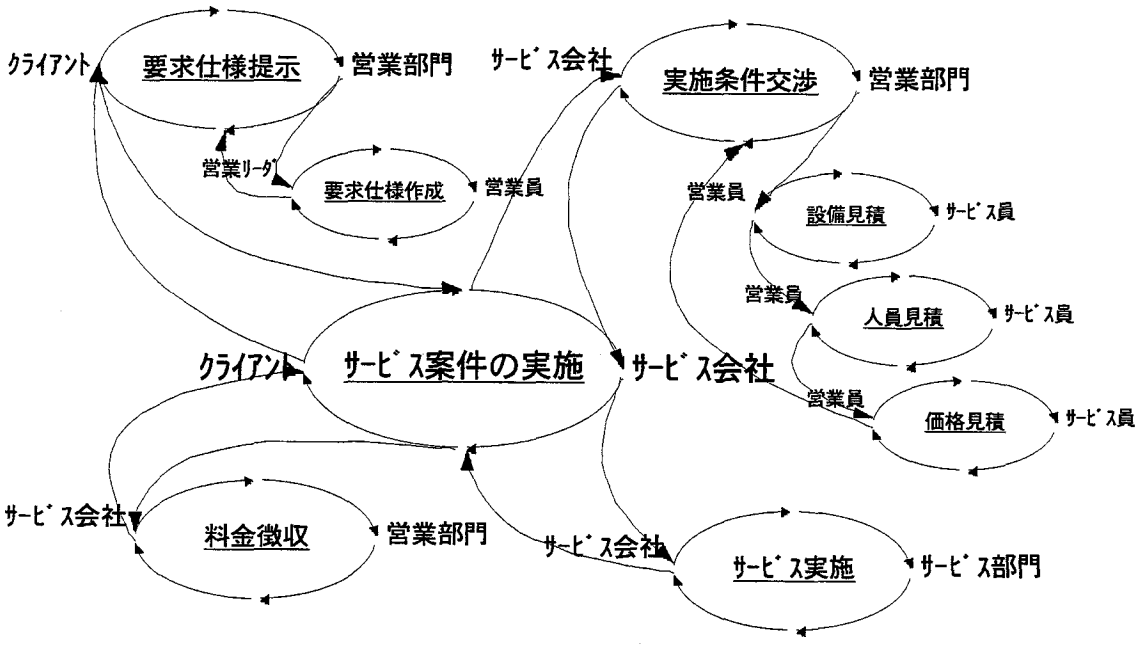


図 4.10 新ビジネスプロセスモデル

この新しいビジネスプロセスの実装は、前記の設計方法に従って次のように行った。

(i)設計パターンとして抽出した４つの基本タスクー要求仕様提示、実施条件交渉、サービス実施、料金徴収ーに対して、それぞれを支援するアプリケーションプログラム（基本アプリケーション）を開発する。

(ii)要求仕様提示と実施条件交渉の２つのタスクは、更に詳細なサブタスクに分割するため、基本アプリケーションを適用して各サブタスク用のアプリケーションプログラムを開発する。

(iii)基本アプリケーションの適用は、具体的には次のように行った。

・基本アプリケーションで参照、更新するデータを、サブタスク用アプリケーションでも共有する。

・サブタスクの画面処理プログラムは、基本アプリケーションのものをコピーし、その中のデータ項目のうち、不要なものを削除、非表示、アクセス禁止等に変更する。

・タスク実行に必要な種々のメッセージ、関連情報を、上記の画面処理プログラムに添付して表示する。

提案方式によるメンテナンス性向上の効果を示すために、上記の方法で実装されたワークフローシステムに対して環境変化に伴う仕様変更を行い、その工数を従来方式と比較する。ここでは、基本タスクの１つである実施条件交渉タスクのサブタスクー設備見積、人員見積、価格見積に、図 4.11 に示す次の２種類の変更を加える。

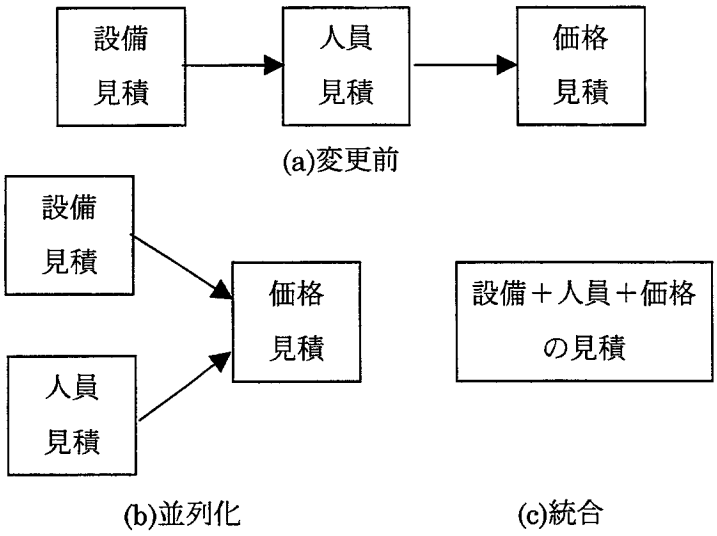


図 4.11　新業務プロセスの変更事例

①タスクの並列化：図 4.10 の交渉状態に属するサブタスクである設備見積、人員見積、価格見積のうち、最初の２つを並列化する。

②タスクの統合：上記の３つのサブタスクを統合して１つのサブタスクとする。

図 4.12 に、実施条件交渉タスクで用いる見積書の基本アプリケーションの例を示す。設備見積、人員見積、価格見積の各サブタスクは設計パターンではないため、それらのタスク分割方法と実行順序は環境に応じて変化する。従って、基本アプリケーションは、設備見積、人員見積、価格見積の実行順序を特定せずに設計する。各サブタスクに応じた個別アプリケーションは、基本アプリケーションをカスタマイズすることにより開発する。すなわち、基本アプリケーションに対して、参照する必要のない項目を非表示にする、更新してはいけない項目を書込み禁止にする、等の変更を行う。

図 4.12　基本アプリケーションの例（見積書）

一方、従来方式では、基本タスクとそれ以外のタスクとの切り分けができないため、通常、図 4.13 に示すように、設備見積、人員見積、価格見積サブタスクごとにアプリケーションを開発する。そして、各アプリケーションの画面レイアウトは、業務担当者の要件にそった個別のものとなる。また、最初に必要な設備を見積り、その後で設備ごとに人員を見積るというサブタスクの順序性を想定した仕様となる。具体的には、図 4.13 の人員見積アプリケーションは、設備ごとに人員を割り当てる形式になっている。

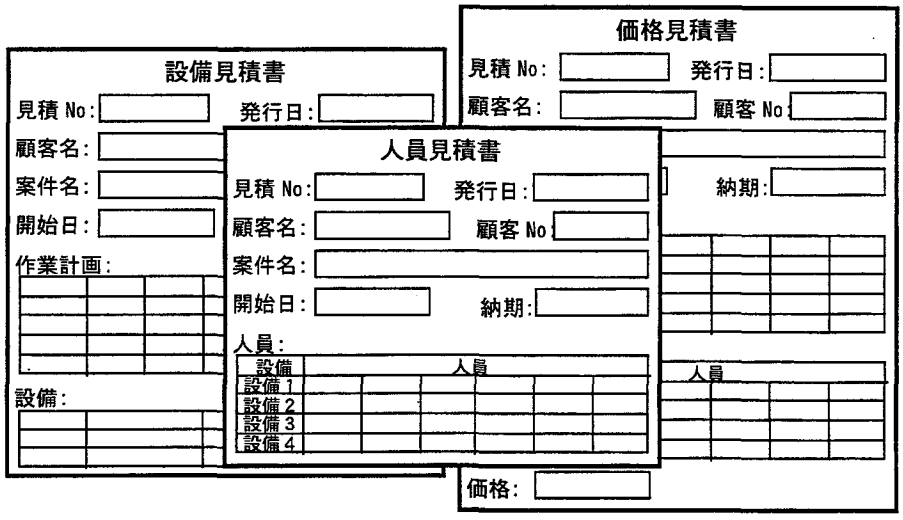


図 4.13　従来方式によるアプリケーションの例（見積書）

以上の前提のもとに、プロセス変更に伴うワークフローシステム仕様の変更内容を表 4.3 示す。まず、タスクの並列化を行った場合について検討する。提案方式では、サブタスクの順序性を前提とせずにアプリケーションを開発するため、逐次的なフローを並列化してもアプリケーションの処理仕様の変更はない。ワークフロー設計仕様を変更するだけである。これに対して、従来方式では、アプリケーションの処理仕様は、設備見積が先、人員見積が後という論理を仮定している。このため、設備見積における設備データ項目、人員見積における人員データ項目の２項目に関して、処理仕様の見直しが必要となる。これは、設備見積、人員見積、価格見積のアプリケーションに属する全データ項目（11 項目）の約 20％の変更となる。

表 4.3　プロセス変更に伴うワークフローシステム仕様の変更内容

#	プロセス変更内容	ワークフローシステム仕様の変更内容	
		提案方式	従来方式
1	並列化	ワークフロー設計仕様の変更のみ必要。アプリケーション処理仕様の変更不要。 (変更項目：3/8)	ワークフロー設計仕様の変更と、設備、人員の各データ項目についてアプリケーション処理仕様の見直し。 (変更項目：3/8+2/11)
2	統合	ワークフロー設計仕様の変更のみ必要。作業計画、設備、人員、価格の各データ項目属性の見直しが必要であるが、工数として無視しうる。 (変更項目：3/8)	ワークフロー設計仕様の変更と、作業計画、設備、人員、価格の各データ項目のマージが必要。 (変更項目：3/8+4/11)

次に、タスクの統合を行った場合を考える。提案方式では、タスクの分割方法は前提とせず、全てのサブタスクのアプリケーションを、基本アプリケーションを適用して開発する。設備見積、人員見積、価格見積の各アプリケーションの処理仕様はほとんど同一であり、部分的にあるデータ項目が非表示であったり、書込み禁止であったりするだけである。従って、アプリケーションの統合に伴い、作業計画、設備、人員、価格の各データ項目を見直し、ワークフロー設計仕様を変更するだけでよい。これに対して、従来方式では、設備見積、人員見積、価格見積のサブタスクごとに個別にアプリケーションを開発する。このため、タスク統合の際には、各アプリケーションに分散している作業計画、設備、人員、価格の各データ項目と処理を１つのアプリケーションに合併する必要がある。

タスクの並列化、統合に伴うワークフローシステム仕様の変更内容が表 4.3 に示す通りであり、ワークフロー設計仕様の１ノードを変更する工数と、アプリケーション処理仕様の１項目を変更する工数が同じであると仮定した場合、提案方式と従来方式のメンテナンス工数の比較は図 4.14 のようになる。

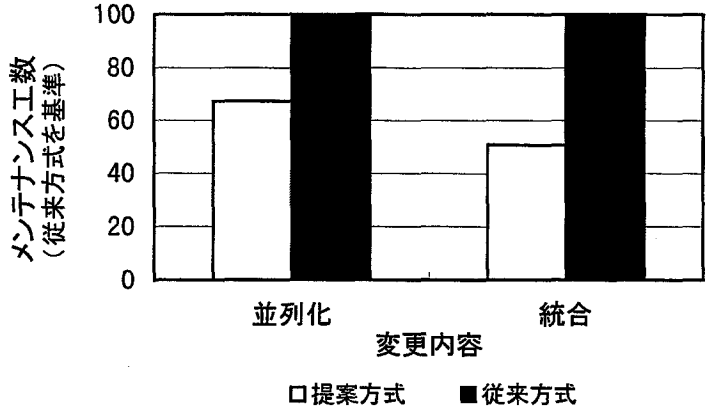


図 4.14　提案方式によるメンテナンス工数向上の効果

図4.14によれば、プロセスの並列化、統合のどちらの場合も提案方式の効果が大きいが、特に、複数のアプリケーションを１つにまとめる必要があるプロセス統合に関して極めて大きな効果があることがわかる。

以上により、設計パターンを適用したワークフローシステム構築技法のメンテナンス性向上の効果が明らかになった。

4.5 結言

本章では、ワークフローシステムのメンテナンス性を向上するためのシステム構築技法を検討した。ワークフローシステムがビジネス環境の変化に柔軟に追従するためには、システム仕様の安定した部分と変化しやすい部分とを明確に分離することが重要である。この課題を実現するために、設計パターンという新しい概念を提案した。ここでは、ビジネスプロセスにおけるタスクの委託階層に着目し、最上位レベルの基本タスクの順序関係をプロセスの安定構造であると考え、プロセスパターンとして抽出した。そして、基本タスクの入出力情報をもとに、業務処理の必要十分な情報構造である情報パターンを抽出した。これらの設計パターンに基づいて、ワークフローシステムの安定仕様を得るための構築手順を提案した。最後に、提案方式を受注管理業務のワークフローシステム構築に適用し、メンテナンス性向上の効果を示した。

提案方式は（株）日立製作所のサービス商品であるワークフロー業務分析設計サービスの標準開発手順に採用され、企業内の販売管理、商品開発、融資稟議、購買申請、等の業務を対象として、現在までに 10 システム以上の構築実績をもっている。今後の課題は、企業内部のビジネスプロセスだけでなく、企業間にまたがる電子商取引プロセスまで範囲を拡大することである。このような全く新しいビジネスプロセスの設計パターンを抽出するためには、世界規模のネットワークを利用して商取引のスタイルがどのように変わるかを十分に検討することが必要となる。

第５章　間接部門の申請業務を対象としたワークフロー業務テンプレートの開発

5.1　緒言

本章では、ワークフローシステム構築の生産性を向上するための、ワークフロー業務テンプレートについて検討する。第４章で提案した設計パターンが多くの企業に共通するビジネスプロセスと情報の標準仕様を示すことに着目し、この考え方に基づいて種々の業務に対応した標準仕様を抽出し、ワークフロー業務テンプレートとしてライブラリ化する。そして、ワークフローシステム構築の際には、ワークフロー業務テンプレートを個別仕様に合わせてカスタマイズすることにより設計工数を低減する。さらに、設計結果を実装するための支援ツールを提案する[76,77]。

ワークフローを適用したシステムを構築する場合、通常、対象とする業務の設計に大きな工数を要する。それは次の２つの理由による。第１に、企業のビジネスプロセスの多くは人間が主体となって行われており、その仕様の詳細が文書化されていることはまれである。また、仮に文書化されているとしても、それは個々の部門単位に分断されたものであり、ビジネスプロセス全体にわたる手順ではない。従って、ビジネスプロセスの全貌を掴むためには、複数の異なる部門に所属する業務担当者と面談して、ビジネスプロセス仕様の断片を統合する必要がある。第２に、ビジネスプロセスは、局所的な改善や変更が長年の間繰返され、多くの場合、全体としての仕様に矛盾、冗長、不足、等がある。このため、実運用に耐えるようにプロセス仕様の見直しが必要となる。さらに、ワークフローツールの適用により業務の並列化、自動化等のメリットを得るために、新しいビジネスプロセスを設計することが必要となる。以上に述べた理由により、業務設計は、ワークフローシステムの構築に慣れていない人にとって極めて敷居の高いものとなっている。

業務ごとにビジネスプロセス仕様と情報仕様の標準形が明らかになっていれば、業務設計は比較的容易なものとなる[39,78]。それらの標準仕様に基づいて、対象プロセスに固有な仕様だけをカスタマイズすれば済むからである。この場合、標準仕様の割合が個別仕様のそれに比べて多いほど、上流設計に要する工数を低減できる。このような業務ごとのビジネスプロセスと情報の標準仕様をワークフロー業務テンプレートとよぶことにする。

本章では、まず、ワークフローシステム構築における問題点を指摘し、それを解決するためのワークフロー業務テンプレートの要件を明らかにする。次に、ワークフロー業務テンプレート開発のための最重要課題であるビジネスプロセス標準仕様の抽出方法を提案する。そして、本方法に基づいて開発したワークフロー業務テンプレートと実装支援ツールを紹介する。最後に、ワークフロー業務テンプレートの適用により、システム構築に要する工数を大幅に低減できることを事例により示す。

5.2 ワークフロー業務テンプレートの要件

ワークフローシステム構築の生産性に関する問題点を述べ、この問題を解決するためのワークフロー業務テンプレートの要件を明らかにする。

5.2.1 従来方式と問題点

ワークフローシステムの構築は、第4章の図4.2で述べたように、業務設計、システム設計、実装という3フェーズに沿って行われる。図5.1にワークフローシステム構築手順の図を再掲する。

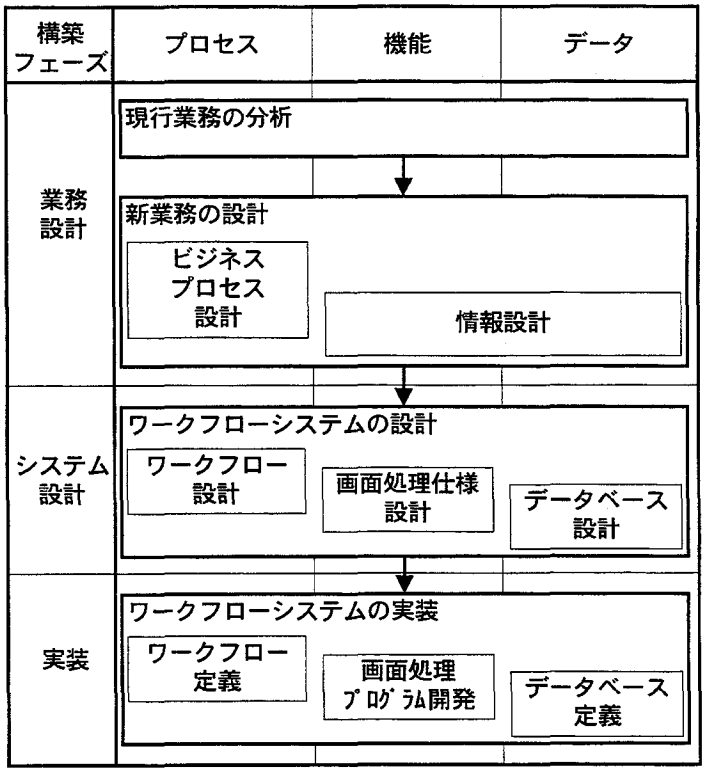


図5.1 ワークフローシステム構築手順

ここで注意すべきことは、ワークフロー適用の目的は対象とするビジネスプロセス全体の実行管理であり、会計処理、顧客管理処理、等のような個々の業務機能の自動化ではないという点である。新規に開発するアプリケーションのほとんどは、タスクの実行指示と実行結果入力のための画面処理、および、既存システムとのインタフェースとなるデータベースの入出力処理であり、比較的容易に開発できる機能である。従って、ワークフローシステム構築に必要な工数の多くが、ビジネスプロセス設計を中心とする業務設計フェーズで費やされる。

前節で述べたように、業務設計に多くの工数を要する理由は次の２点である。

(1) 対象業務のビジネスプロセスと情報に関する仕様が文書化されておらず、複数部署に常識、慣習という形態で散在している。これらを抽出し、一貫した仕様として整理する必要がある。

(2) ビジネスプロセスと情報に関する仕様の中に、本質的で遵守しなければならないものと流動的で変更できるものが混在している。業務改善／改革のためにはこれらを区別しなければならず、対象業務の規則、組織、資源、等に関する知識が必要となる。

このような理由から、業務設計は極めて専門的な要素が高く、対象業務の専門家、業務分析者、情報技術者の三者が協力して、多くの工数を費やして実施せざるを得ない[79]。それを示す事例を図 5.2 に示す。どの例も、業務設計が全体工数の 20%～40%を占めている。これらの例では、業務分析者と情報技術者が２～３人のチームを編成し、業務の専門家に面談する方法をとり、２ヶ月～６ヶ月を費やして業務設計を行っている。

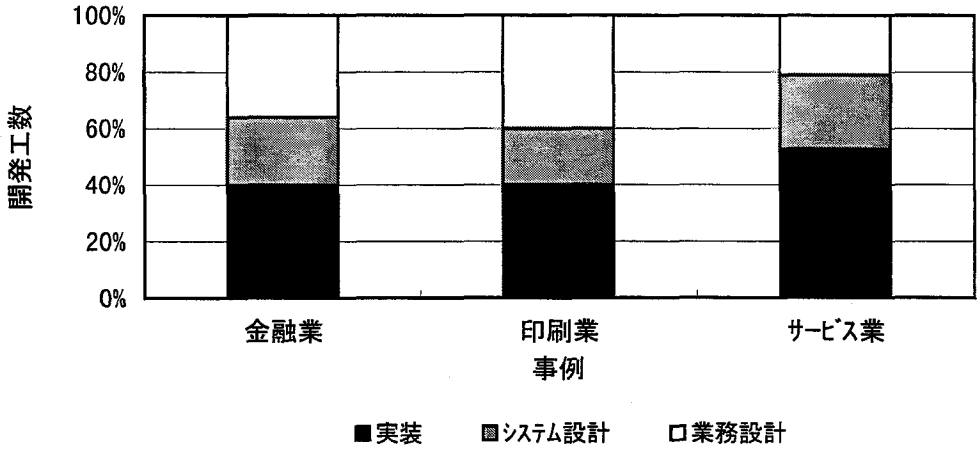


図 5.2 業務設計に要する工数の事例

5.2.2 ワークフロー業務テンプレートの要件

ワークフロー業務テンプレートの目的は、業務設計に高度な専門技術と膨大な工数を要するという問題を解決することにある。そのために、業務設計の成果物であるビジネスプロセス仕様と情報仕様を再利用するというアプローチをとる。すなわち、予め業務ごとにビジネスプロセスと情報の標準仕様を抽出しライブラリ化しておき、ワークフローシステムを構築する際には、その標準仕様を個別業務に合わせてカスタマイズする。さらに開発工数を削減するために、これらの標準仕様に基づいて、ワークフロー定義、画面処理仕様、データベース定義というシステム設計の成果物も事前に作成しておき再利用する。

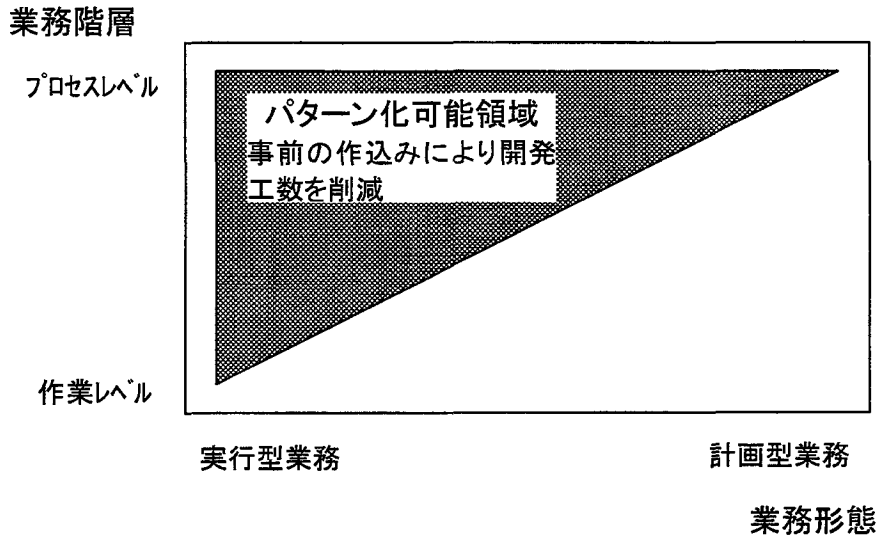


図 5.3 ビジネスプロセス仕様パターン化の可能性

ところで、ビジネス分野で行われている各種業務のビジネスプロセス仕様は、その階層レベルを適切に選べば、多くの企業に共通なパターンを見出すことが可能である[39,80,81]。ここで、ビジネスプロセス仕様の階層レベルには、販売、生産、調達、等の概略手順を規定するプロセスレベルのものから、見積り品目決定、在庫確認、価格設定、等の詳細手順を示す作業レベルのものまで存在する。上位レベルのビジネスプロセス仕様ほど汎用的なパターンを見出すことが容易である。一方、業務の実行形態には、予め定義されたビジネスプロセス仕様を効率よく実現することを使命とする実行型の業務と、ビジネスプロセス仕様の全てを事前に決められず実行中にそれを設計することが必要な計画型のものがある。前者には販売管理、生産管理、人事管理、等があり、後者には経営計画、生産計画、販売計画、等がある。一般に、実行型業務は計画型業務に比べてパターン化しやすい。図 5.3

は、業務階層と業務形態に応じたパターン化可能領域の分布を示したものである。

以上の議論に基づいてワークフロー業務テンプレートの要件をまとめると、図 5.4 に示す通りとなる。すなわち、企業内業務の中からビジネスプロセス仕様がパターン化しやすい実行型業務を選択し、各業務の標準的なビジネスプロセス仕様を抽出してライブラリ化する。また、ビジネスプロセスの標準仕様が決まれば、そこで入出力される情報の構造も標準化できる。さらに、このような業務設計の標準仕様に基づいて、システム設計の成果物を作成しライブラリ化する。ワークフローシステムを構築する際には、これらの標準仕様を個別業務に合わせてカスタマイズすることにより設計を行う。さらに、設計結果を実装するための支援環境も提供する。以下に、詳細な要件を図 5.4 に従って説明する。

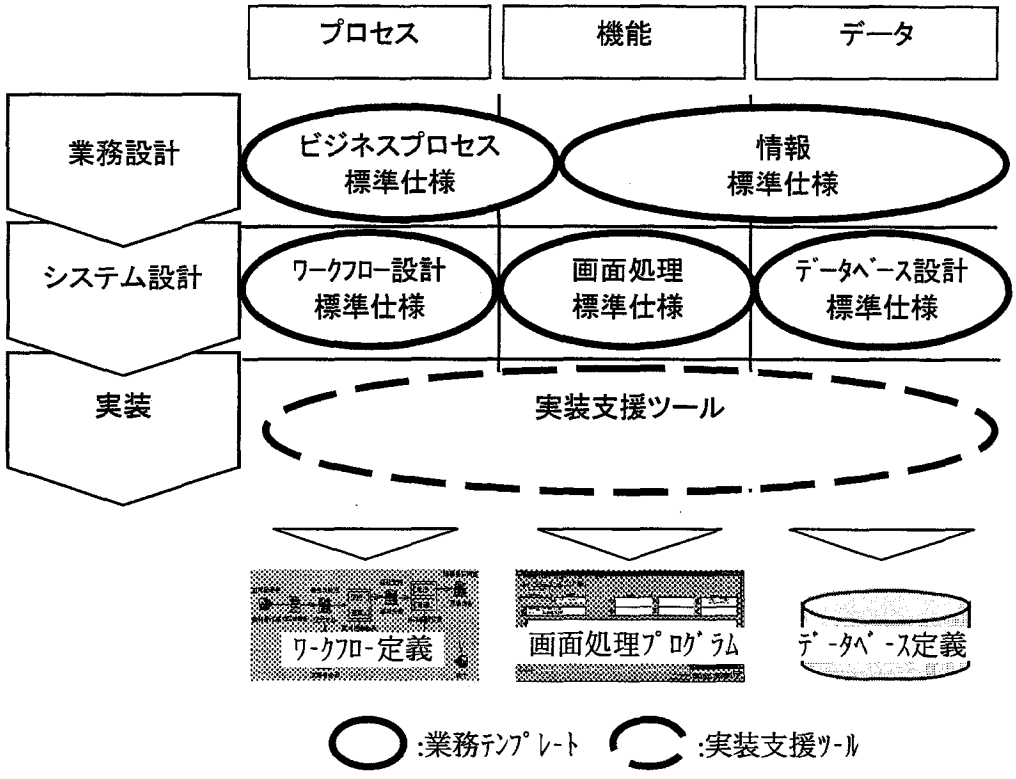


図 5.4　ワークフロー業務テンプレートの要件

(1) 業務設計の標準仕様

ワークフローシステムを構築するための業務設計の標準仕様として、各業務ごとに、ビジネスプロセス仕様と情報仕様を抽出しライブラリ化しておく。前述したように、ビジネスプロセス標準仕様の抽出が最大の課題であり、それが明らかになれば、情報仕様は画面

処理とデータベース処理に関するものとして比較的単純に抽出できる。

(2) システム設計の標準仕様

上記の業務設計の標準仕様に基づいて、ワークフローツール、プログラム言語、データベースにより実装することを前提に、システム設計の標準仕様を開発する。具体的には、ワークフロー設計、画面処理仕様設計、データベース設計を事前に行い、その成果物を業務設計の標準仕様と対応づけてライブラリ化する。

(3) 実装支援ツール

上記の標準システム仕様のカスタマイズ結果を、実装につなげるための支援環境を提供する。これは、市販の定義ツールを極力利用する。

5.3 ワークフロー業務テンプレート

5.3.1 ビジネスプロセス標準仕様抽出の考え方

ワークフロー業務テンプレートを実現するための最大の課題は、ビジネスプロセス標準仕様の抽出である。この標準仕様を明らかにする方法として考えられる１つの案は、業務ごとに複数の企業のビジネスプロセスを調査し、それらの共通仕様を抽出するものである。この方法は堅実な方法ではあるが、多くの時間と工数が必要となる。特に、ビジネスプロセス標準仕様をライブラリ化するためには、大量の業務分析者を動員する組織的なアプローチが必要となり実現性が低い。最小の時間と工数でワークフロー業務テンプレートを実現するためには、業務ごとに複数のビジネスプロセス仕様を比較する方法でなく、単一のビジネスプロセスから標準仕様を抽出する方法が必要となる。

前章で提案した設計パターンは、環境変化の影響を受け難いビジネスプロセスと情報の安定構造であり、ワークフローシステムのメンテナンス性の向上を実現するものである。このように時間的に変わり難い仕様は、一般に、空間的にも変化し難く[78]、多くの企業で共通して利用できるビジネスプロセスと情報の標準仕様となり得る。従って、設計パターンの抽出方法に基づいて、ビジネスプロセス標準仕様と情報標準仕様を導くことができる。この方法によれば、業務ごとに複数のビジネスプロセスを比較して共通仕様を見出すのでなく、単一のビジネスプロセスを分析するだけで標準仕様を得ることができる。従って、ワークフロー業務テンプレートを実用的な時間と工数で実現することができる。

5.3.2 間接部門申請業務の設計パターン

設計パターンを適用してワークフロー業務テンプレートを開発するためには、5.2 節で

述べたように、企業内の各種業務の中から定型性が高い実行型業務を選ぶ必要がある。本章では、その典型的なものとして間接部門の申請業務を採用する。本業務は、企業内の各部門から総務、経理等の間接部門に対して依頼された各種の申請案件を処理するものである[82]。依頼部門と間接部門のただ２つの部門が関わり、依頼、審査・承認、実行、結果通知という単純な手順で処理される。申請業務は、その機能により、経理、人事、購買、総務の４つに細分され、各業務の概略の処理手順は表 5.1 に示す通りである。これらのうち、経理、人事、購買系業務は、各々、会計、就業、購買に関する法律と社内規則に基づく極めて定型性の高い業務であり、帳票の書式が厳密に規定され、専用の計算機システムにより自動的に処理される部分が多い。一方、総務系業務は、それらと比較して幾分非定型的な要素があり、帳票の書式は概略しか規定されておらず、主に人間の裁量で処理される。従って、ワークフロー業務テンプレートの対象業務としては、経理、人事、購買系業務が適している。

表 5.1　間接部門申請業務の設計パターン

#	業務分類	概略処理手順	特徴
1	経理系 申請業務 (入出金、旅費)	依頼 → 審査承認 → 経理ｼｽﾃﾑ → 結果通知	各種取引の精算申請。会計規則に基づく定型帳票と経理システムで処理。
2	人事系 申請業務 (勤怠、申告)	依頼 → 審査承認 → 人事ｼｽﾃﾑ → 結果通知	勤怠管理と管理情報更新。就業規則に基づく定型帳票と人事システムで処理。
3	購買系 申請業務 (物品、ｻｰﾋﾞｽ)	依頼 → 審査承認 → 購買ｼｽﾃﾑ → 結果通知	消耗品、労働力等の購入。購買規則に基づく定型帳票と購買システムで処理。
4	総務系 申請業務 (依頼、伺い)	依頼 → 審査承認 → ｻｰﾋﾞｽ実施 → 結果通知	従業員に対する各種サービスの実施。非定型帳票を人間の裁量で処理。

ワークフロー業務テンプレートを開発するためには、上記の業務分類をさらに詳細化し帳票レベルまで細分する必要がある。例えば、購買系申請業務に属する物品購買申請の設計パターンは図 5.5 と図 5.6 に示すようになる。図 5.5 はプロセスパターンであり、購買

依頼、購買承認、発注、検収というタスクにより構成される。図 5.6 は情報パターンであり、注文依頼伝票、審査用伝票、発注伝票、検収伝票を中心に、組織、予算費目、商品、取引先等の実体が相互に関連をもつ。これらの設計パターンを標準仕様として利用する。

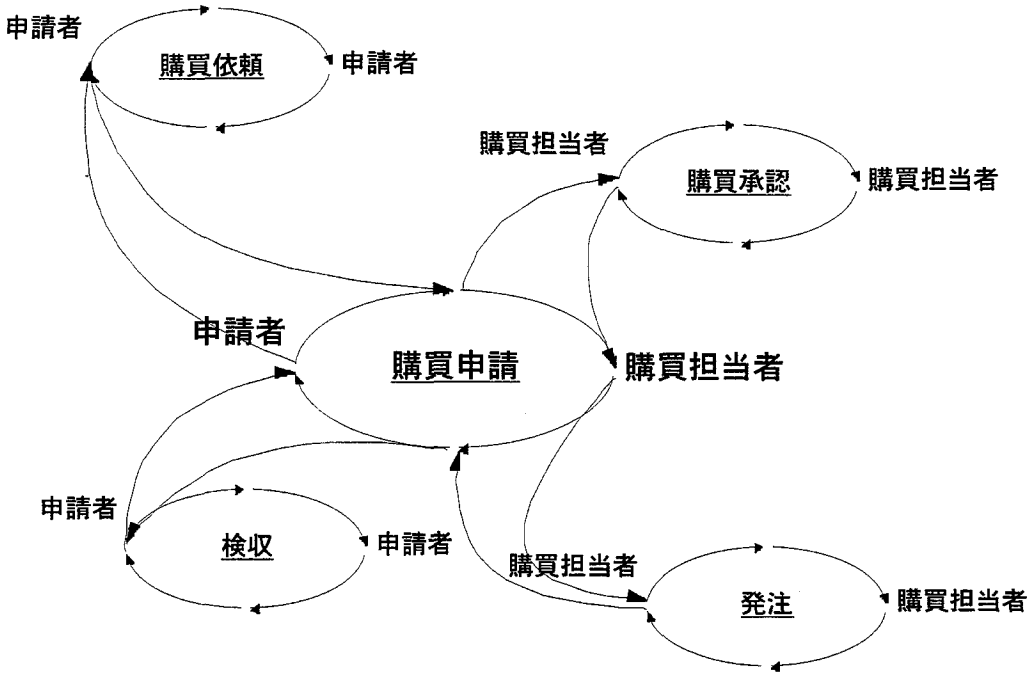


図 5.5 物品購買申請のプロセスパターン

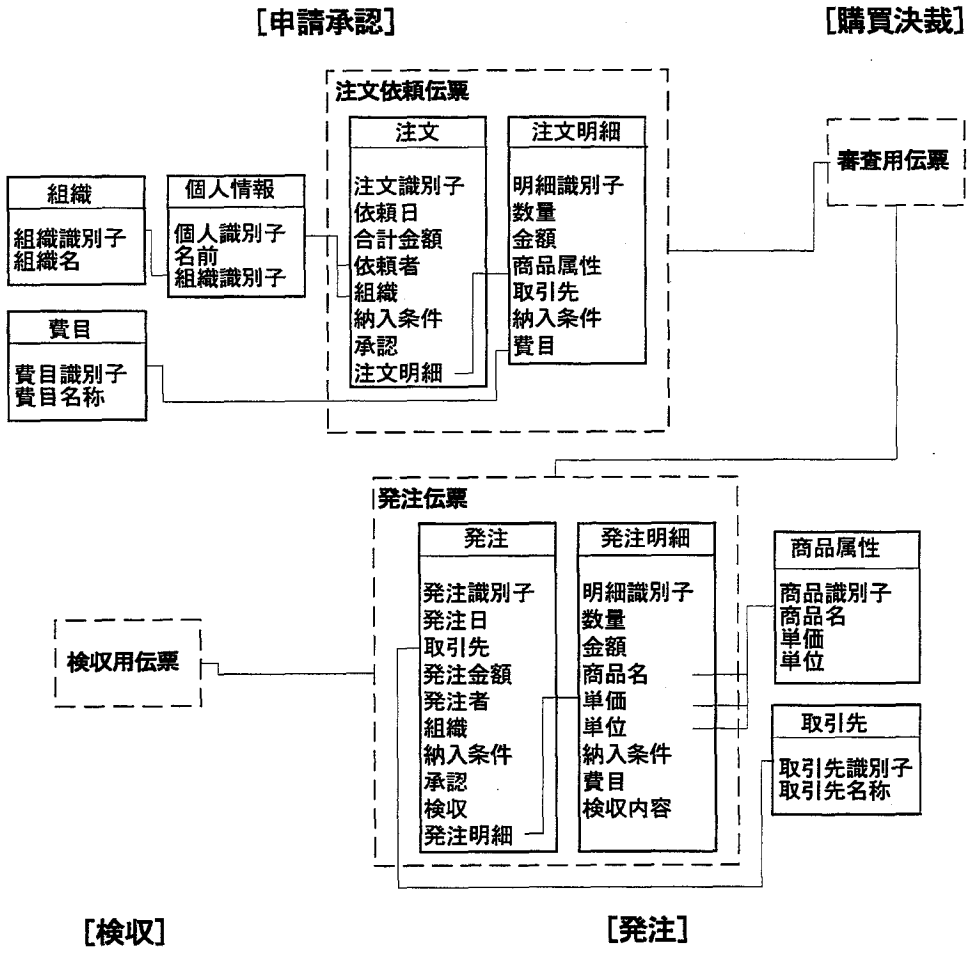


図 5.6 物品購買申請の情報パターン

5.3.3 ワークフロー業務テンプレートと実装支援ツールの全体構成

前項で示した間接部門の申請業務の設計パターンに基づいて、ワークフロー業務テンプレートおよび実装ツールを開発する。図 5.7 にそれらの全体構成を示す。

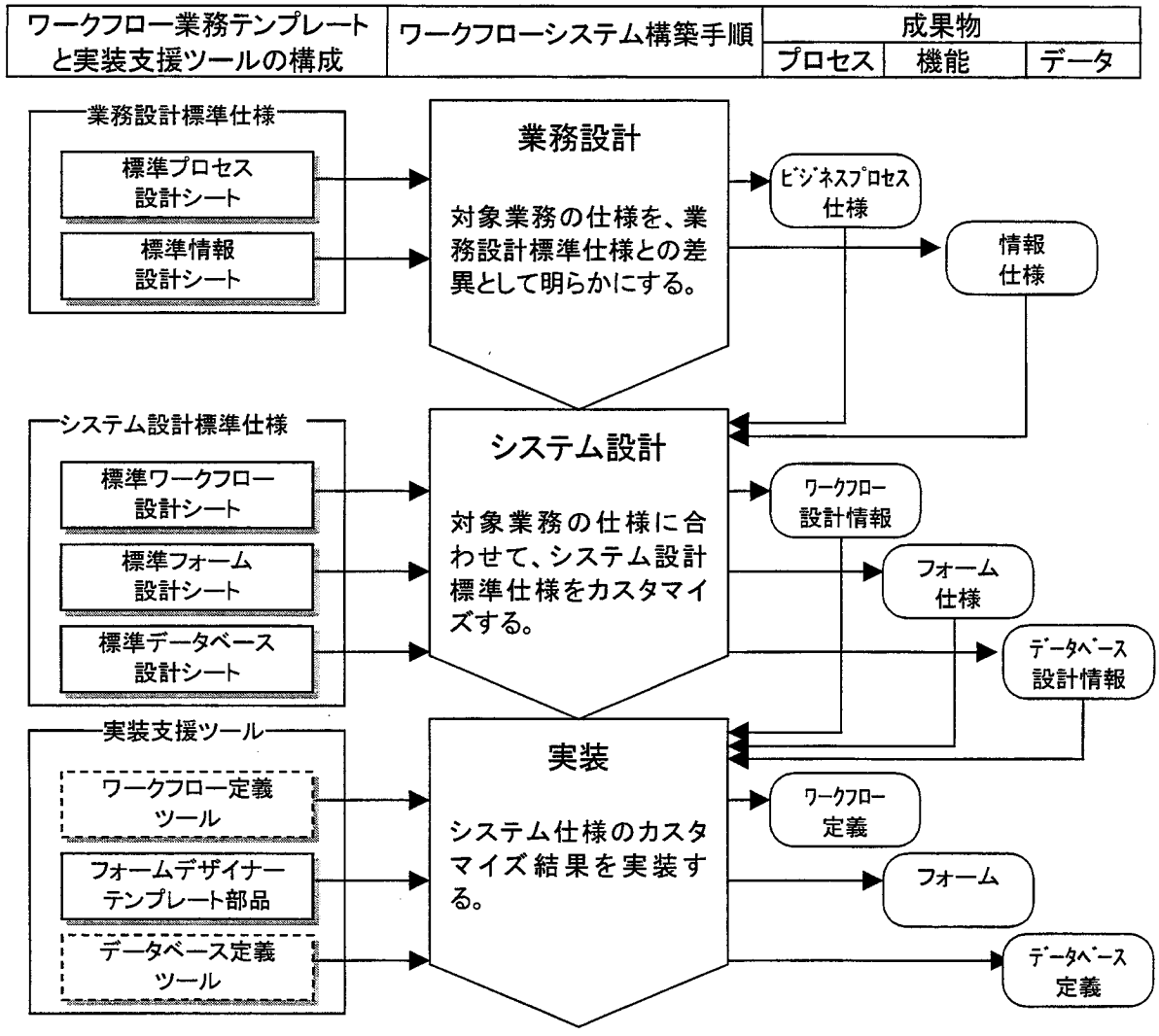


図 5.7 ワークフロー業務テンプレートと実装支援ツールの全体構成

ワークフロー業務テンプレートは、図 5.7 に示すように、業務設計の標準仕様とシステム設計の標準仕様により構成する。業務設計の標準仕様は設計パターンに基づくものであり、プロセスパターンにより得られる標準プロセス設計シートと、情報パターンにより得られる標準情報設計シートからなる。業務設計フェーズにおいては、対象業務の仕様をこれらの標準仕様との差異として明らかにし、ビジネスプロセス仕様と情報仕様を設計する。システム設計の標準仕様は業務設計の標準仕様に基づくものであり、標準ワークフロー設

計シート、標準フォーム設計シート、標準データベース設計シートの３つにより構成する。システム設計フェーズにおいては、対象業務仕様に合わせてシステム設計標準仕様をカスタマイズする。

実装支援ツールとしては、画面処理仕様を実装するために、フォームデザイナーとテンプレート部品を開発する。フォームデザイナーは、標準ワークフロー設計シートのカスタマイズにより得られたワークフロー設計情報と、標準フォーム設計シートのカスタマイズにより得られたフォーム仕様を入力とする。そして、ワークフローから呼び出されて画面処理を行うプログラム（フォーム）を自動生成する。テンプレート部品は、フォームデザイナーで生成されたフォームに対して、さらに個別の機能を追加する場合に利用する。なお、ワークフロー定義のためにはワークフローツールに標準装備の定義ツールを利用し、データベース定義のためには市販のデータベース定義ツールを利用する。

5.3.4 ワークフロー業務テンプレート

ワークフロー業務テンプレートの詳細を、表 5.2 に従って、以下に説明する。

表 5.2 ワークフロー業務テンプレートの構成物

#	構成物		説明
1	業務設計の標準仕様	標準プロセス設計シート	業務毎のビジネスプロセス標準仕様。ビジネスプロセス設計は、この設計シートをベースに必要に応じてカスタマイズを行う。
		標準情報設計シート	業務毎の情報標準仕様。情報設計は、この設計シートをベースに必要に応じてカスタマイズを行う。
2	システム設計の標準仕様	標準ワークフロー設計シート	ビジネスプロセス標準仕様に基づく、ワークフローツールの標準定義情報。この設計シートをベースに、ノードの追加／削除、ロールの変更、分岐条件の変更等を行う。
		標準フォーム設計シート	情報標準仕様に基づく、フォーム（画面処理プログラム）の標準定義情報。この設計シートをベースに、フォームのデータ項目の追加／削除、属性の変更等を行う。
		標準データベース設計シート	情報標準仕様に基づく、データベースの標準定義情報。この設計シートをベースに、データベースのデータ項目の追加／削除、属性の変更等を行う。

(1) 業務設計の標準仕様

業務設計の標準仕様を示す２つの設計シートは、業務ごとに設計パターンを記述したものである。図 5.7 に示すように、対象業務の仕様を、これら標準仕様との差異として把握するためのワークシートとして利用する。

a. 標準プロセス設計シート：この設計シートは、業務ごとに図 5.5 に示す標準的なプロセスパターンを記述したものである。個々のプロセス設計を行うためのワークシートとして利用し、必要に応じて、担当者レベルの詳細タスクを追加する。
b. 標準情報設計シート：この設計シートは、業務ごとに図 5.6 に示す標準的な情報パターンを記述したものである。個々の情報設計を行うためのワークシートとして利用し、必要に応じて、実体関連の変更、データ項目の追加／削除等を行なう。

(2) システム設計の標準仕様

システム設計の標準仕様を示す３つの設計シートは、図 5.7 に示すように、業務設計の標準仕様に基づいて明らかになったビジネスプロセス仕様と情報仕様を、情報システムとして実現するためのワークシートである。図 5.8 にそれらの事例を示す。

a. 標準ワークフロー設計シート：この設計シートには、ビジネスプロセス標準仕様に基づいてワークフロー定義情報が既に設計されている。ワークフロー設計時は、それを個別仕様に合わせてカスタマイズする。ワークフロー定義情報には、ノード名称、起動アプリケーション名称、ロール名称、ユーザ属性、分岐条件、等がある。この設計結果は、ワークフロー定義ツールの入力情報となる。
b. 標準フォーム設計シート：この設計シートには、ワークフローの中で画面処理を行うプログラム（フォーム）の処理仕様が、情報標準仕様に基づいて既に設計されている。フォームの設計時は、それを個別仕様に合わせてカスタマイズする。フォーム処理仕様には、データ項目の名称、属性、値域、等がある。この設計結果は、後述のフォームデザイナーの入力情報となる。
c. 標準データベース設計シート：この設計シートには、標準情報仕様に基づいてデータベース定義情報が既に設計されている。データベース設計時は、それを個別仕様に合わせてカスタマイズする。データベース定義情報には、各種の台帳、定数テーブルのデータ項目の名称、属性、値域、等がある。この設計結果は、データベース定義ツールの入力情報となる。

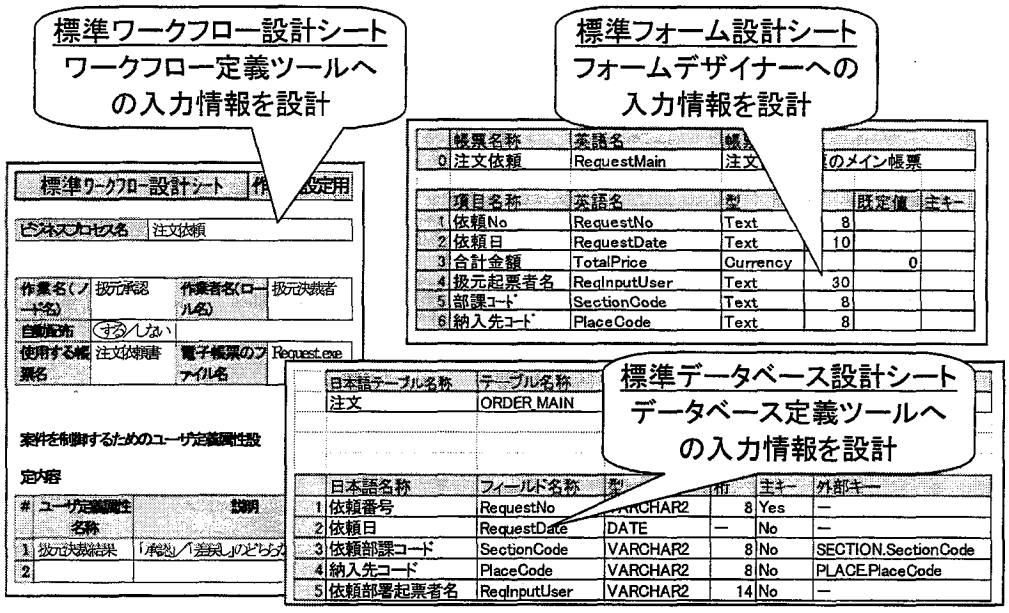


図 5.8　システム設計標準仕様の事例

5.3.5　実装支援ツール

前項で示したワークフロー業務テンプレートを利用して明らかにしたワークフロー、フォーム、データベースに関する設計結果を、ワークフローシステムとして実装するための支援ツールを提案する。図 5.7 に示すように、プロセスとデータの実装のためには、各々、市販のワークフロー定義ツールとデータベース定義ツールを適用する。従って、実装支援ツールは、表 5.3 に示すように、フォームデザイナーとテンプレート部品により構成する。以下、実装支援ツールの詳細を説明する。

表 5.3　実装支援ツールの構成物

#	構成物	説明
1	フォームデザイナー	標準フォーム設計シートを用いて一連のデータ項目の属性を定義すれば、それらのデータ項目フィールドとワークフロー連携処理を組み込んだフォームプログラムを自動生成するツール。
2	テンプレート部品	フォームのデータ項目処理とワークフロー連携処理を実装するためのソフトウェア部品ライブラリ。一部はフォーム自動生成の際の標準処理を実装するために利用し、その他は、標準処理以外のさらに高度な処理を開発する際に利用する。

(1) フォームデザイナー

フォームとは、既に述べたように、ワークフローツールから呼び出される画面処理プログラムであり、ワークフローユーザとの間で、作業指示データの画面出力、作業結果データの画面入力、等を行う。また、ワークフローツールに対して、予め定義された次の担当者への作業データ（案件）の送信指示を行う。さらに、データベースに格納された関連データの検索、作業結果のデータベース登録を行う。これらのフォーム機能のうち、開発上のボトルネックとなるのはワークフローツールとの連携機能である。通常のアプリケーション開発者は、汎用のプログラミング言語により画面処理やデータベース入出力処理を開発することは得意であるが、ワークフローツール固有のインタフェース言語を使ってワークフロー連携処理を開発することには不慣れであることが多い。

フォームデザイナーはこの問題を解決するものであり、図 5.9 に示すように、標準フォーム設計シートを用いて一連のデータ項目の属性を定義すれば、それらのデータ項目フィールドとワークフロー連携処理を組み込んだフォームプログラムを自動生成する。この際、各データ項目の値域を指定すれば、範囲チェック処理を自動的に生成する。また、メニュー選択用の定数テーブルを指定すれば、メニュー選択形式のデータ入力処理を自動生成する。さらに、フォームプログラムに生成したデータ項目の値を、ワークフローの分岐制御の判定値として指定することも可能である。

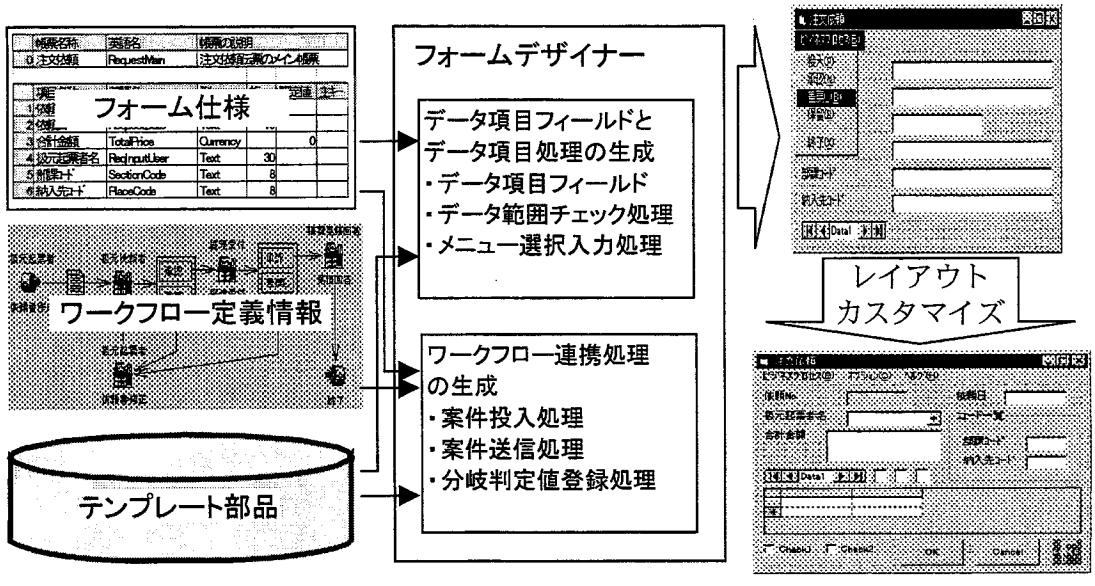


図 5.9　フォームデザイナーの機能

以上のように自動生成されたフォームは、データ項目フィールドが定義順に並べられた

だけの暫定的なものであるため、実用に適するようにレイアウトを設計する必要がある。また、フォームデザイナーで自動生成する標準的な処理以外の、さらに高度な処理を開発したい場合は、次に述べるテンプレート部品を利用することにより工数を低減できる。

なお、フォームデザイナーはMicrosoft社のVisual Basicにより開発されWindows上で稼動する。自動生成するフォームは Visual Basic のものである。ワークフローツールは（株）日立製作所の Groupmax Workflow を、データベースは Microsoft 社の Access を前提とする。

(2) テンプレート部品

テンプレート部品は、フォームのデータ項目処理とワークフロー連携処理を実装するためのソフトウェア部品ライブラリである。フォームデザイナの説明で述べたように、それらの一部はフォーム自動生成の際の標準処理を実装するために利用する。表 5.4 の＊印のついているものがそれに該当する。その他は、標準処理以外のさらに高度な処理を開発する際に利用する。

表 5.4　テンプレート部品

#	種別	部品名	#	種別	部品名
1	データ項目処理	データ範囲チェック* メニュー選択入力* ノード別処理制御	2	ワークフロー連携処理	案件投入* 案件送信* 分岐判定値登録* ログイン／アウト ユーザ情報取得 案件位置判定 添付情報追加／削除

以下に、フォーム自動生成の際に標準で実装されない部品の機能を説明する。データ項目処理部品のノード別処理制御は、ノード別にデータ項目の属性、処理を変更したい場合に利用する。例えば、審査ノードにおいてデータ項目の一部を書込み禁止とする際に用いる。ワークフロー連携処理部品のログイン／アウト処理は、ワークフローツール標準のログイン／アウト画面でなくユーザ固有のものを開発する場合に利用する。また、ユーザ情報取得処理は、ログイン名称に基づいてユーザの名前、所属等の属性を取得してフォームのデータ項目に自動設定するために用いる。案件位置判定処理は、当該案件が指定されたノード位置に存在するか否かを判定するもので、前記のノード別処理制御とともに用いる。添付情報追加／削除処理は、フォームに対してドキュメント、イメージファイル等を添付

するために利用する。

5.4 ワークフローシステム構築事例

上記のワークフロー業務テンプレートおよび実装支援ツールに基づくシステム構築手順を、図 5.7 に従って説明する。

(1) 業務設計フェーズ

まず、対象業務の仕様を調査する。この際、業務設計の標準仕様に基づいて、ビジネスプロセスと情報に関する標準的な仕様を把握し、それらと対象業務との差異に注目する。そして、この相違点がシステム構築後も必要であるか否かをエンドユーザと議論することにより、プロセス改善を行う。その結果、なお相違点が残る場合には、標準仕様のカスタマイズが必要となる。

図 5.10 は、物品購買申請の標準プロセス設計シートに基づくプロセス設計の事例であり、実線はビジネスプロセス標準仕様を構成するタスクを示し、破線は対象プロセス向けに追加したタスクを示す。ここでは、購買部門の承認に加えて経理部門の承認を追加し、購買品が予算の範囲内にあるか否かのチェックを行うこととした。

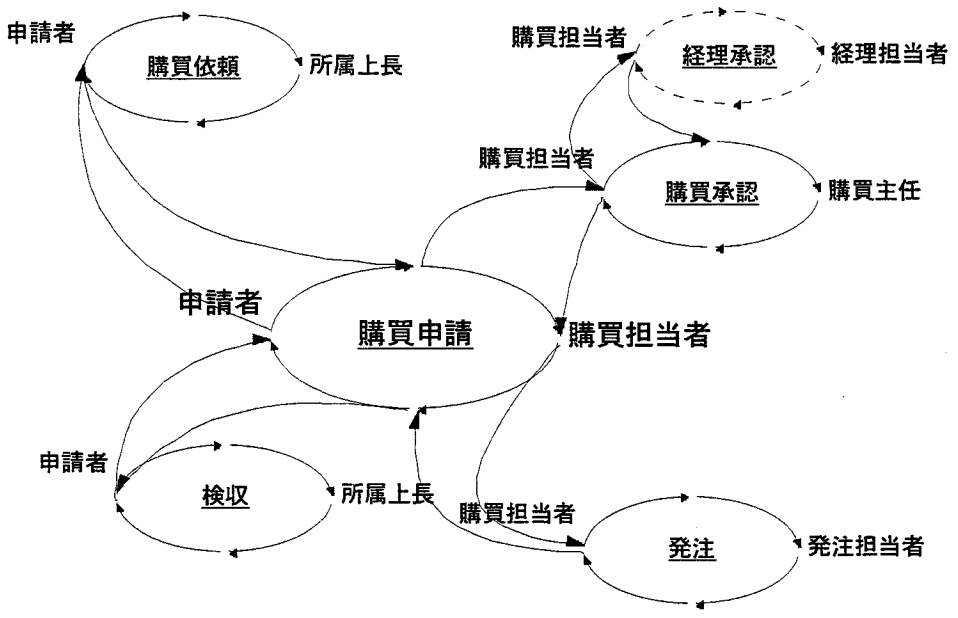


図 5.10 標準プロセス設計シートに基づくプロセス設計事例

図 5.11 は、物品購買申請の標準情報設計シートに基づく情報設計の事例であり、実線のボックスは情報標準仕様を構成する実体を示し、破線のボックスは対象プロセス向けに追加した実体を示す。図 5.10 において経理承認タスクを追加したことに伴い、予算ファイルを追加し、注文依頼伝票に予算チェックのためのデータ項目（予算、予算チェック）を追加した。また、商品ごとに取引先を限定するための対応テーブルを追加した。

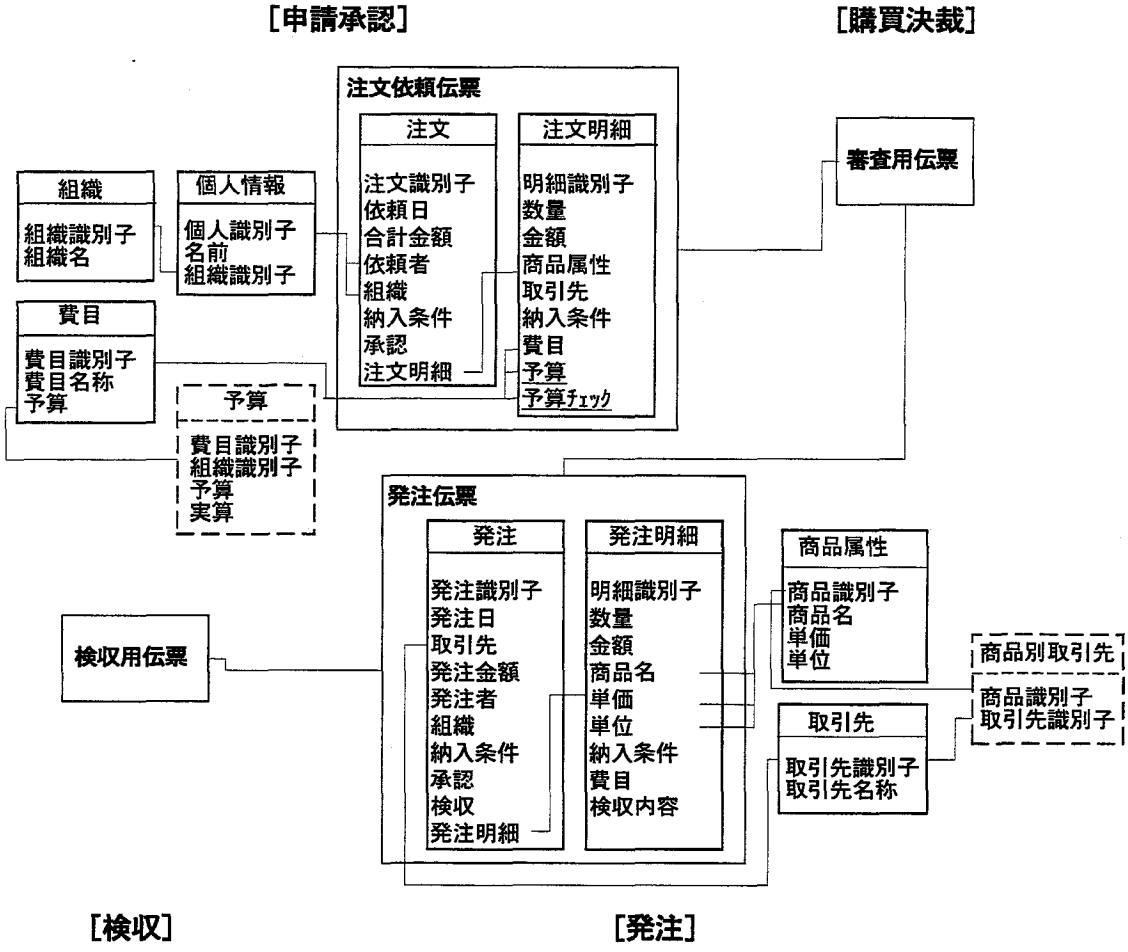


図 5.11　標準情報設計シートに基づく情報設計事例

(2) システム設計フェーズ

業務設計で明らかにした、対象業務のビジネスプロセス仕様と情報仕様に基づいて、ワークフロー設計、フォーム設計、データベース設計を行う。

ワークフロー設計では、標準ワークフロー設計シートに対して、新たに経理承認タスクに関する定義情報を追加する。フォーム設計では、標準フォーム設計シートの注文依頼伝

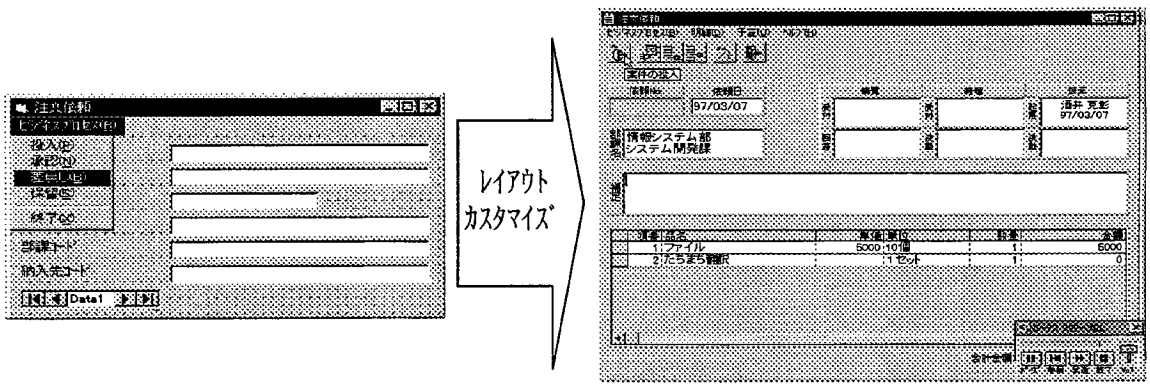
票明細テーブルに対して、新たに２つのデータ項目（予算と予算チェック）を追加する。データベース設計では、標準データベース設計シートに対して、新たに予算テーブルと商品別取引先テーブルを追加作成する。

(3) 実装フェーズ

設計結果に基づいて、ワークフローシステムとして実装を行う。まず、ワークフロー設計で明らかになった定義情報を、ワークフロー定義ツールを用いて設定する。同様に、データベース設計で明らかになった定義情報を、データベースツールを用いて設定する。

フォーム設計で明らかにした処理仕様は、フォームデザイナーを利用して実装する。本ツールによれば、フォームを構成するデータ項目の各種属性（データ型、桁数、初期値、値域、等）を設定することにより、データ項目フィールドとワークフロー連携処理を組み込んだフォームプログラムを自動生成できる。

ただし、フォームデザイナーにより自動生成したフォームはデータ項目が定義順に羅列してあるだけなので、レイアウトをカスタマイズする必要がある。図 5.12 は、自動生成されたフォームと、それにレイアウトのカスタマイズを行った後のフォームの例を示したものである。次に、テンプレート部品を利用して、購買品の価格が予算金額内か否かのチェック処理と、購買品を扱う取引先のメニュー表示処理を作成する。



＜自動生成されたフォーム＞　　＜カスタマイズ後のフォーム＞

図 5.12　カスタマイズ前後のフォームの事例

5.5　効果

ワークフロー業務テンプレートの効果を示すために、前節で示した物品購買申請業務の

ワークフローシステムの構築に際して、ワークフロー業務テンプレートを適用した場合とそうでない場合について開発状況を比較する。前者は実際にシステム構築を行って工数を計測したもの、後者は机上で見積もったものである。表 5.5 はそれら２つのケースの開発状況を、構築フェーズごとに比較したものである。

表 5.5 ワークフローシステム開発状況の比較

#	構築フェーズ	提案方式（ワークフロー業務テンプレートと実装支援ツールを適用）	従来方式
1	業務設計	業務設計標準仕様のカスタマイズ。 ・１つのタスクの追加 ・２つの実体の追加	プロセス、情報仕様をゼロの状態から設計。 ・５つのタスクの順序関係 ・15 個の実体の関連
2	システム設計	システム設計標準仕様のカスタマイズ。 ・ワークフロー仕様の１つのノード追加と順序関係の変更 ・２つのプログラム仕様の変更 ・２つデータベーステーブルの追加	業務設計の結果に基づいて、ワークフロー定義情報（ノード数：５）、フォームおよびデータベースのデータ項目（項目数：56）、プログラム（項目数：36）を全て設計。
3	実装	フォームレイアウトのデザインと、上記のシステム設計結果の実装。	上記のシステム設計結果の実装。

業務設計フェーズでは、既に述べたように、ビジネスプロセス設計に特に工数を要する。提案方式によれば、ビジネスプロセスの標準仕様は既に抽出されており、５つのタスクのうち経理承認タスクだけを検討すればよいため１／５以下の工数で済む。一方、システム設計フェーズでは、フォームデータ項目のプログラム設計に工数を要する。提案方式によれば、36 個のフォームデータ項目のうち、２項目のプログラム仕様を検討すればよいため１／18 以下の工数で済む。実装フェーズでは、フォームレイアウトのデザインとプログラムの実装を行う。フォームのレイアウトデザインの工数は従来方式と同じであるが、プログラムの実装は変更した２項目だけでよいため１／18 以下の工数で済む。

業務設計、システム設計、実装に要する工数の割合が、5.2節で示した事例に従うと仮定した場合、提案方式による開発工数低減の効果は、図5.13に示す通りとなる。この結果によれば、業務設計とシステム設計フェーズでは、標準仕様を個別仕様に合わせてカスタマイズすればよいため極めて高い効果があることがわかる。また、実装においても、自動生成されたフォームのレイアウトを業務仕様に合わせてデザインし、必要最小限の処理プロ

グラムをテンプレート部品の適用により付加するだけで済む。さらに、付随的な効果として、標準的なシステム仕様と実装例が業務テンプレートという形で明らかになっていることにより、ワークフローシステムのエンドユーザと開発者との間で、業務仕様の個別性を標準仕様のカスタマイズの範囲で吸収するという共通意識が生まれる。従って、業務設計とシステム設計においては、業務テンプレートとの相違点だけに焦点を当てて検討を行うことができる。

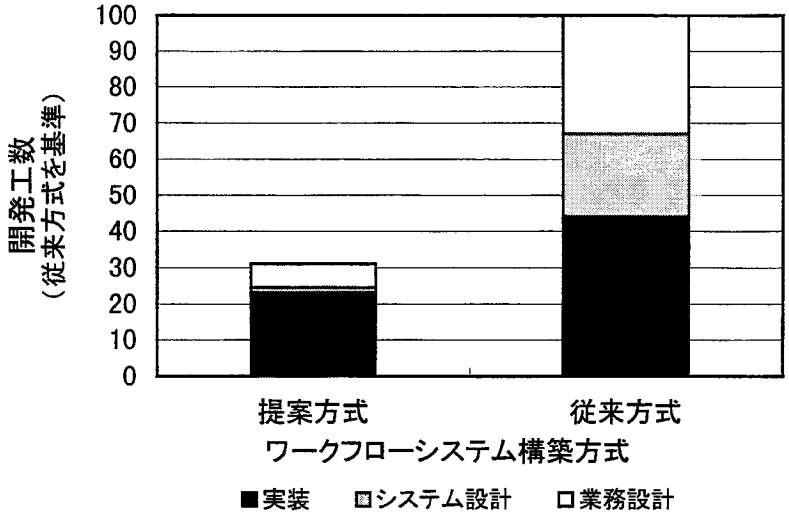


図5.13　提案方式による開発工数低減の効果

以上に述べたように、ワークフロー業務テンプレートの適用によりワークフローシステムの開発工数を大幅に削減できる。

5.6　結言

本章では、ワークフローシステム構築のボトルネックである業務設計の工数を削減するために、その成果物であるビジネスプロセス仕様と情報仕様の再利用について検討した。第４章で提案した設計パターンを適用することにより、ビジネスプロセスと情報に関する標準仕様を抽出し、ワークフロー業務テンプレートとしてライブラリ化した。これにより、業務設計とシステム設計を標準仕様のカスタマイズで行えるようにした。さらに、その設計結果を実装するための支援ツールを提案した。最後に、ワークフロー業務テンプレートと実装支援ツールの適用により、従来方式に比べて大幅な工数削減の効果があることを事

例により示した。ワークフロー業務テンプレートのような標準仕様ライブラリの有効性を示すためには、開発工数低減の効果とともに、どのくらい適用範囲が広いかという再利用性の評価が必要である。これは、今後、適用実績をあげて評価データを収集した後の検討課題であると考えている。

現在までに、本章で例として示した物品購買申請の他に、勤怠管理と入出金業務のワークフロー業務テンプレートを開発している。これらのテンプレートに基づき、間接部門の申請業務全般のワークフローシステム化を短期間かつ低工数で実施するサービスを計画中である。今後の課題として、販売管理、生産管理等の定型性の高い業務を対象として、ワークフロー業務テンプレートの品揃えを拡大する。また、アプリケーションプログラムの開発工数をより低減するために、フォームデザイナーとテンプレート部品の機能を充実させる。

第6章　結　論

本論文では、計算機工学をはじめとする種々の技術分野の分散処理システムを対象として、ジョブ管理方式の最適化と柔軟性に関して議論した。そして、計算機工学分野の段階的ヒューリスティックス適用とその評価方法、および、経営工学分野のプロセス安定構造の抽出に基づくシステム構築方法に関する研究成果を、以下の５章に分けて述べた。

第１章では、分散処理システムのジョブ管理方式を定義しその目的が最適化と柔軟性のトレードオフであることを述べた。そして、最適性と柔軟性という２つの分類軸に従って課題を整理することにより、計算機工学、生産工学、経営工学の各分野におけるジョブ管理方式の研究動向を明らかにした。その結果に基づき、本研究の課題として、①相互に影響する複数の特性をもつジョブのスケジューリング問題の解決、および、②環境変化に強い柔軟なジョブ管理システムの実現を設定した。そして、前者の課題のためには「段階的ヒューリスティックス適用とその評価」という方針をとり、後者の課題のためには「プロセス安定構造の抽出に基づくシステム構築」という方針をとった。

第２章では、計算機入出力装置の一種であるカートリッジ磁気テープライブラリ装置の共有リソースの競合回避を題材として、段階的にヒューリスティックスを適用するジョブ管理方式の開発方法を検討した。ここでは、テープ運搬ジョブの割当問題を、①最適ドライブ装置の選択、②最適アクセッサへのジョブ割当、の２つのサブ問題に分割した。そして、最適ドライブ選択においてライブラリの左半分の領域または右半分領域だけで実施できる運搬ジョブを生成し、最適アクセッサ割当において右アクセッサは右領域、左アクセッサは左領域のジョブを割り当てる方針をとることにより、２つのサブ問題を相互作用の少ない単純なものとし、ヒューリスティックスの適用を可能とした。その結果、アクセッサ間の干渉を大幅に低減でき、応答性能とスループットを向上することができた。

第３章では、複数のメインリソースとサブリソースにより構成される典型的な分散処理システムを対象として、ヒューリスティックなジョブスケジューリング方式の評価方法を検討した。ここでは、スケジューリング方式ごとにシステム状態とジョブ処理性能の静的な関数（性能関数）を導き、これを確率ペトリネットモデルに組込んで動的な性能評価を行った。この方法により、スケジューリング方式の詳細なアルゴリズムをモデル化する必要がなくなったため、解空間の規模を大幅に縮小することができ、シミュレーションより

も少ない工数でヒューリスティックス方式の評価を行うことができた。そして、第２章で論じたカートリッジ磁気テープライブラリ装置のジョブ管理方式の評価を事例として、本評価方式の有効性を示した。

第４章では、ワークフローシステムのメンテナンス性を向上するためのシステム構築技法を検討した。ここでは、環境変化の影響を受け難いビジネスプロセスの安定構造と、そこで入出力される情報の構造を示す設計パターンという新しい概念を提案した。設計パターンを適用することにより、ワークフローシステムの共通仕様と個別仕様を明確に分離することができるため、仕様変更に伴うシステムの改造範囲を最小限に留めることができた。

第５章では、ワークフローシステム構築のボトルネックである業務設計の工数を削減するために、その成果物であるビジネスプロセス仕様と情報仕様の再利用について検討した。第４章で提案した設計パターンを適用することにより、ビジネスプロセスと情報に関する標準仕様を抽出し、ワークフロー業務テンプレートとしてライブラリ化した。これにより、業務設計とシステム設計を標準仕様のカスタマイズで行えるようにした。さらに、その設計結果を実装するための支援ツールを提案した。その結果、従来方式に比べて大幅な工数低減を実現した。

最後に、今後の課題についてまとめる。

(1) 「段階的ヒューリスティックス適用とその評価」の適用範囲の拡大と体系化

本論文では、計算機入出力装置の一種であるカートリッジ磁気テープライブラリ装置を事例として、本方法の効果を示した。しかし、ヒューリスティックス適用のための問題分割方法、および、性能関数の設定方法はまだ体系的な手順が整理されておらず、解析者の直観に負うところが多い。今後、マルチプロセッサ構成のディスク制御装置、計算機ネットワークシステム等、種々の分散計算機システムに適用し、本方法の有効性と限界を見極めるとともに、設計、評価手順としての体系化を検討する。

(2) 「プロセス安定構造の抽出に基づくシステム構築」の適用範囲の拡大と体系化、および、設計パターンライブラリの充実

本方法の適用実績は、既に、受注管理、生産管理、商品開発、融資稟議、不動産実査、購買申請、勤怠管理、旅費精算、等、十数個にのぼる。今後、これらの適用範囲をさらに拡大して設計パターンの抽出方法を体系化するとともに、設計パターンライブラリの質と量を充実させる。

(3) 分野横断的なジョブ管理方式の適用

本論文において、計算機工学、生産工学、経営工学等を含む広い分野を想定して、分散

処理システムのジョブ管理方式を検討した。この背景には、ある分野で有効な方法を他の分野の問題解決にも活用したいという意図がある。例えば、経営工学の分野では、まだ人間が主体で業務を行っているが、今後、ワークフロー、電子メール等のオフィスインフラストラクチャーの普及に伴い、単純なジョブの自動処理化が進む可能性が高い。そうなると、経営工学のジョブ管理に生産工学や計算機工学における成果を活用できるはずである。このよう分野横断的なアプローチを推進するような、モデルと方法論の開発を行っていく。

謝　辞

本研究の全過程を通じて、終始懇切なるご指導とご鞭撻と格別のご配慮を賜りました、大阪大学大学院 工学研究科 情報システム工学専攻 薦田憲久 教授に慎んで深謝の意を捧げます。

本研究をまとめるにあたり、貴重なお時間を割いて頂き、丁寧なるご教示を賜りました、大阪大学大学院 工学研究科 情報システム工学専攻 西尾章治郎 教授に慎んで深謝の意を捧げます。

大学院博士後期課程において、情報システム工学全般に関して懇切なるご指導とご助言を賜りました、大阪大学大学院 工学研究科 情報システム工学専攻 村上孝三 教授、白川功 教授、藤岡弘 教授、下條真司 教授に慎んで深謝の意を捧げます。

本研究を遂行するにあたり、格別のご指導を頂き、有益なるご助言を賜りました、大阪大学大学院 工学研究科 情報システム工学専攻 大川剛直 助教授、一階良知 助手に心から御礼申し上げます。

筆者が大阪大学大学院 工学研究科 情報システム工学専攻 博士後期課程に在学する上で、終始暖かいご指導とご鞭撻を賜った、(株) 日立製作所 システム開発本部 本部長の正坊地邦典氏、元主任技師（現四日市大学教授）の都島功博士、第２部長の津田道夫氏、第５部長の楠崎哲生氏、元業務サポート室室長の元田陵作氏、現室長の土屋匡右氏に厚く御礼申し上げます。

第２章および第３章の、カートリッジ磁気テープライブラリ制御方式の研究の機会を与えていただくとともに有益な示唆をいただいた、(株)日立製作所 システム開発研究所 元所長の堂免信義氏、元主管研究長の故三巻達夫博士、元主管研究員（現大阪工業大学教授）の中尾和夫博士、元企画室長（現桃山学院大学教授）の明石吉三博士、主管研究長の舩橋誠寿博士に心から御礼申し上げます。また、共同研究者として本研究の全般にわたって様々なご討論、ご助言をいただいた、(株) 日立製作所 システム開発研究所 元主任研究員（現神奈川大学助教授）の栗原謙三博士、第６部研究員の里山愛氏に心から御礼申し上げます。さらに、本研究を遂行する上でご協力をいただいた、(株) 日立製作所 ストレージシステム事業部 元主管技師長の宮崎道生氏、システム設計部 元主任技師の土井隆氏、製品開発本部 主任技師の小菅稔氏に深く感謝申し上げます。

第４章および第５章の、ワークフローシステムの構築技法の研究にあたって様々なご教授、ご討論、ご支援をいただいた、（株）日立製作所 情報システム統括営業本部 主管技師長の村田文也氏、同本部 主任技師の関根章司氏に心から御礼申し上げます。また、共同研究者として本研究の全般にわたってご協力をいただいた、（株）日立製作所 システム開発本部 第２部 技師の趙延悦氏、大村義秀氏、同部の尾越昌子氏、日立情報システム（株）主任技師の竹内成明氏、同社の酒井克彰氏に心から感謝します。さらに、本研究を遂行する上でご協力をいただいた、（株）日立製作所 ストレージシステム事業部 システム応用設計部長の西山淳氏、同部技師の上野浩一氏、販売企画部長の飛田正士氏、製品企画部 部長代理の漢那宗丈氏、ソフトウェア事業本部 第２オープンプラットフォーム設計部 副部長の鬼頭政義氏、同部 主任技師の玉樹正人氏、 Action Technology Incorporated の Mr. Philip Hallstein に深く感謝いたします。

筆者の所属している薦田研究室の小野田仙一氏、圓田哲也氏には、ワークフローシステム構築技法およびビジネスプロセスモデルの解析方法の研究にあたり共同研究者としてご協力をいただきました。また、同研究室の諸兄には本研究の遂行にあたり様々な面でご協力をいただきました。ここに記して深く御礼申し上げます。

最後に、いつも笑顔で見守ってくれた妻 悦子に感謝いたします。

参考文献

1 Winograd, T. and Flores, F.: Understanding Computers and Cognition – A New Foundation for Design, Addison-Wesley (1985)

2 Marsan, M. A., Balbo, G. and Conte, G.: Performance Models of Multiprocessor Systems, MIT Press (1986)

3 人見勝人: 生産システム工学（第２版），共立出版 (1998)

4 Silva, M.: Petri Nets and Flexible Manufacturing, G. Rozenberg(ed.), Advance in Petri Nets 1989, Lecture Notes in Computer Science No. 424, Springer-Verlag, pp. 374/417 (1990)

5 椎塚久雄: 実例ペトリネット, コロナ社 (1992)

6 西垣通 (編): 組織とグループウェア, NTT 出版 (1992)

7 西垣通 (編): 思想としてのパソコン, NTT 出版 (1997)

8 Davenport, T. H.: Process Innovation - Reengineering Work through Information Technology, Harvard Business School Press (1993)

9 Hammer, M. and Champy, J.: Reengineering the Corporation – A Manifesto for Business Revolution, HarperBusiness (1993)

10 Hammer, M.: Beyond Reengineering – How the Process-Centered Organization Is Changing, HarperBusiness (1996)

11 Lawrence, P. (ed.): Workflow Handbook 1997, John Wiley & Sons (1997)

12 Fischer, L. (ed.): New Tools for New Times - The Workflow Paradigm Second Edition, Future Strategies Inc. (1994)

13 戸田保一, 飯島淳一, 速水治夫, 堀内正博: ワークフロー － ビジネスプロセスの変革に向けて, 日科技連 (1998)

14 Jablonski, S. and Bussler, C.: Workflow Management – Modeling Concepts, Architecture and Implementation, International Thomson Computer Press (1996)

15 Kobielus, J. G.: Workflow Strategies, IDG Books Worldwide (1997)

16 松田光明, ワークフローオートメーション実践入門, 同友館 (1996)

17 Jackson, M. and Twaddle, G.: Business Process Implementation – Building Workflow Systems, Addison-Wesley (1997)

18 Myers, G. J.: Composite/Structured Design, Litton Education Publishing (1978)

19 Lockyer, K. and Gordon, J.: Project Management and Project Network Techniques, Pitman Publishing (1996)

20 Filman, R. E. and Friedman, D. P.: Coordinated Computing – Tools and Techniques for Distributed Software, McGraw-Hill (1984)

21 解説 ハードディスク装置 サーバ向けはFC-AL対応にPC向けは10Gバイト超へ, 日経エレクトロニクス, 1998.1.12, No. 707, pp. 27/38 (1998)

22 Gelenbe, E. I. and Mitrani: Analysis and Synthesis of Computer Systems, Academic Press (1980)

23 Coffman, E. G. and Denning, P. J.: Operating Systems Theory, Prentice-Hall (1973)

24 橋田温, 川島幸之助: 待ち行列ネットワークモデルによる計算機システムの性能評価, 情報処理, Vol. 21, No. 7, pp. 743/750 (1980)

25 山本彰, 坪井俊明, 北嶋弘行, 難波龍雄, 土井隆: カートリッジ型MTにおける予測制御型負荷均衡アルゴリズムとその評価, 情報処理学会 オペレーティングシステム研究会報告 42-2 (1989.2.24)

26 離散事象システム研究専門委員会（編）: ペトリネットとその応用, 計測自動制御学会 (1992)

27 Hatono, I., Yamagata, K. and Tamura, H.: Modeling of FMS under Uncertainty Using Stochastic Petri Nets, in Proc. of The Third International Workshop of Petri Nets and Performance Models, Kyoto, Japan, pp. 122/129 (1989.12)

28 Hatono, I., Yamagata, K. and Tamura, H.: Modeling and On–Line Scheduling of Flexible Manufacturing Systems Using Stochastic Petri Nets, IEEE Trans. on Software Eng., Vol. 17, No. 2, pp. 126/132 (1991)

29 Marca, D. A. and McGowan, C. L.: IDEF0/SADT Business Process and Enterprise Modeling, Electric Solutions Corporation (1986)

30 Jacobson, I., Ericsson, M. and Jacobson, A.: The Object Advantage – Business Process Reengineering with Object Technology, Addison-Wesley (1996)

31 Buhr, R. J. and Casselman, R. S.: Use Case Maps for Object-Oriented Systems, Prentice-Hall (1996)

32 DeMarco, T.: Structured Analysis and System Specification, Prentice-Hall (1979)

33 鯉沼章: フローチャートによる事務分析, 日刊工業新聞社 (1987)

34 小野田仙一, 小林隆, 尾越昌子, 湯本真樹, 薦田憲久: 資材注文依頼業務のワークフロ

ーについてのターンアラウンドタイム分析, 電気学会論文誌, Vol. 118-C, No. 4, pp. 585/591 (1998.4)

35 Onoda, S., Kobayashi, T. and Komoda, N.: Bottleneck Detection Analysis for Workflow Improvement, in Proc. of 1997 IEEE International Conference on System, Man, and Cybernetics, Orlando, Florida, USA, pp. 3331/3336 (1997.10)

36 Maruta, T., Onoda, S., Ikkai, Y., Kobayashi, T. and Komoda, N.: A Deadlock Detection Algorithm for Business Processes Workflow Models, in Proc. of 1998 IEEE International Conference on System, Man, and Cybernetics, San Diego, California, USA , pp. 611/616 (1998.10)

37 Onoda, S., Ikkai, Y., Kobayashi, T. and Komoda, N.: Definition of Deadlock Patterns for Business Processes Workflow Models, in Proc. of the Thirty-Second Hawaii International Conference on System Sciences (HICSS-32), Wailea, Hawaii, USA (1999.1) (to appear)

38 Lang, K., Taumann, W. and Bodendorf, F.: Business Process Reengineering with Reusable Reference Process Building Blocks, Scholz-Reiter, B. and Stickel, E. (eds.), Business Process Modeling, Springer-Verlag, pp. 265/290 (1996)

39 Curran, T. and Keller, G.: SAP R/3 Business Blueprint – Understanding the Business Process Reference Model, Prentice-Hall (1998)

40 Taylor, D. A.: Business Engineering with Object Technology, John Wiley & Sons (1995)

41 Taylor, D. A.: Object-Oriented Information Systems – Planning and Implementation, John Wiley & Sons (1992)

42 Malone, T. W. and Crowston, K.: The Interdisciplinary Study of Coordination, ACM Computer Surveys, Vol. 26, No. 1, pp. 87/119 (1994)

43 Malone, T. W.: Tools for Inventing Organizations – Toward a Handbook of Organizational Process, http://ccs.mit.edu/CCSWP198/ (1997)

44 Yu, L.: A Coordination-based Approach for Modeling Office Workflow, Scholz-Reiter, B. and Stickel, E. (eds.), Business Process Modeling, Springer-Verlag (1996)

45 小林隆, 栗原謙三, 里山愛, 小菅稔: カートリッジ磁気テープライブラリの制御方式, 情報処理学会論文誌, Vol. 33, No. 5, pp. 611/619 (1992.5)

46 小林隆, 栗原謙三, 里山愛: カートリッジ磁気テープライブラリ装置制御方式の提案, 情報処理学会第 41 回（平成 2 年後期）全国大会, 第 4 分冊, pp. 79/80 (1990.9)

47 里山愛, 小林隆, 栗原謙三: カートリッジ MT ライブラリ装置制御方式のシミュレー

ション評価, 情報処理学会第 42 回（平成 3 年前期）全国大会, 第 4 分冊, pp. 55/56 (1991.3)

48 Conway, R. W.: Theory of Scheduling, Addison-Wesley (1967)

49 茨木俊秀: 組合せ最適化の理論, コロナ社 (1979)

50 Feller, W.: An Introduction to Probability Theory and Its Applications Vol. 1, John Wiley & Sons (1950)

51 Kobayashi, H.: Modeling and Analysis, Addison-Wesley (1978)

52 木瀬洋、塩山忠義: 確率スケジューリング問題について, オペレーションズ・リサーチ, 1987 年 11 月号, pp. 750/757 (1987)

53 平成 3 年度の日立技術の展望, 日立評論, Vol. 73, No. 1, p. 71 (1991)

54 Kobayashi, T. and Kurihara, K.: Performance Analysis of Job Scheduling Policy for Interference Avoidance Using Stochastic Petri Nets, IEICE TRANSACTIONS, Vol. E 74, No. 10, pp. 3144/3151 (1991.10)

55 Kobayashi, T. and Kurihara, K.: Performance Model of Interference Avoidance Policy Using Stochastic Petri Nets, in Proc. of 1991 IEEE International Workshop of Petri Nets and Performance Models, Melbourne, Australia, pp. 125/133 (1991.12)

56 小林隆, 栗原謙三: ジョブの干渉回避スケジューリング方策の確率ペトリネットを適用した性能解析, 電気学会研究会資料（システム・制御研究会）, SC-91-8-13, pp. 1/9 (1991.8)

57 Muntz, R. R.: Queuing Networks – A Critique of the State of the Art and Directions for the Future, ACM Computing Surveys, Vol. 10, pp. 353/359 (1987.9)

58 山崎源治, 逆瀬川浩孝: 生産システム設計のための待ち行列モデル, オペレーションズ・リサーチ, 1988 年 5 月号, pp. 215/225 (1988)

59 上野信行: 製造分野における離散系シミュレーション技術の展開, オペレーションズリサーチ, 1993 年 11 月号, pp. 15/21 (1993)

60 梅田茂樹, 森戸晋: 離散系シミュレーションを取り巻く実態と展開, オペレーションズリサーチ, 1993 年 11 月号, pp. 5/9 (1993)

61 薦田憲久, 大川剛直: システムのモデリングとシミュレーション, 計測自動制御学会 (1995)

62 Molloy, M. K.: Performance Analysis Using Stochastic Petri Nets, IEEE Trans. on Computers, Vol. C-31, No. 9, pp. 913/917, (1982.9)

63 Marsan, M. A., Balbo, G. and Conte, G.: A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems, ACM Trans. on Computer Systems, Vol. 2, No. 2, pp. 93/122 (1984.5)

64 Dugan, J. B., Trivedi, K. S., Geist, R. M. and Nicola, V. F.: Extended Stochastic Petri Nets: Application and Analysis, in Proc. of Performance '84, North-Holland, pp. 507/519 (1984)

65 Murata, T.: Petri Nets: Properties, Analysis and Applications, Proc. of IEEE, Vol. 77, No. 4, pp. 541/580, (1989.4)

66 熊谷貞俊, 薦田憲久: ペトリネットによる離散事象システム論, コロナ社 (1995)

67 Peterson, J. L.: Petri Nets Theory and Modeling of Systems, Prentice-Hall (1981)

68 小林隆, 薦田憲久: 高メンテナンス性を指向したワークフローシステム設計技法, 電気学会論文誌、Vol. 118-C, No. 7, pp. 1130/1137 (1998.7)

69 Kobayashi, T., Ogoshi, S. and Komoda, N.: A Business Process Design Method for Applying Workflow Tools, in Proc. of 1997 IEEE International Conference on System, Man, and Cybernetics, Orlando, Florida, USA, pp. 2314/2319 (1997.10)

70 尾越昌子, 小林隆, 大村義秀: ビジネスプロセスモデルを用いた業務分析設計技法の開発, 情報処理学会第 54 回（平成 9 年前期）全国大会, 第 1 分冊, pp. 383/384 (1997.3)

71 大村義秀, 小林隆, 趙延悦: ビジネスプロセス設計パターンを用いたワークフローシステム設計開発技法の開発, 情報処理学会第 55 回（平成 9 年後期）全国大会, 第 4 分冊, pp. 173/174 (1997.9)

72 薦田憲久: 情報システム工学入門, 朝倉書店 (1994)

73 Winograd, T.: A Language/action Perspective on the Design of Cooperative Work, Human-Computer Interaction, Vol. 3, No. 1, pp. 3/30 (1987)

74 Medina-Mora, R., Winograd, T., Flores, R. and Flores, F.: The Action Workflow Approach to Workflow Management Technology, in Proc. of CSCW 92, pp. 1/10 (1992)

75 Schal, T.: Workflow Management Systems for Process Organizations, Lecture Notes in Computer Science No. 1096, Springer-Verlag (1996)

76 小林隆, 小野田仙一, 薦田憲久: ワークフロー業務テンプレートの開発, 人工知能学会研究会資料（知識ベースシステム研究会）, SIG-KBS-9801, pp. 31/36 (1998.9)

77 Kobayashi, T., Ohmura, Y., Ogoshi, S., Sakai, K. and Komoda, N.: Development of Workflow Template for Administration Process, in Proc. of 1998 IEEE International Conference and Workshop on Engineering of Computer Based Systems, Jerusalem, Israel, pp. 290/297

(1998.4)

78 對馬靖人: ソフトウェア部品, 丸山学芸図書 (1998)

79 Kim, K. H. and Kim, Y. G.: Process Reverse Engineering for BPR – A Form-based Approach, Information & Management No. 33, pp. 187/200 (1998)

80 Porter, M. E.: Competitive Advantage – Creating and Sustaining Superior Performance, Free Press (1985)

81 後藤靖國: 業務知識と情報システム, 講談社出版センター (1997)

82 足立信彦, 川原井智博: 中小企業のための人事·総務·経理フォーム集, かんき出版 (1994)