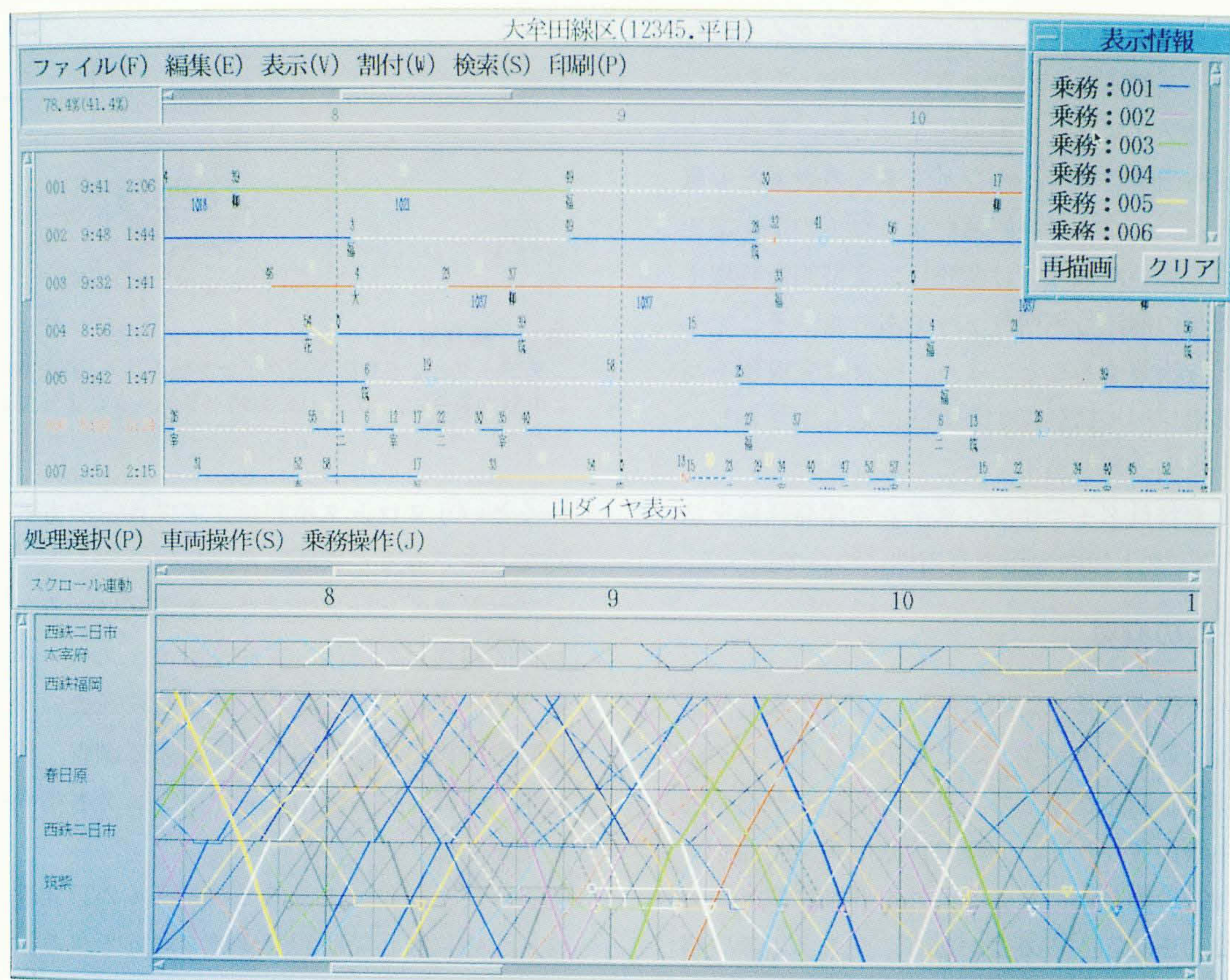


西日本鉄道株式会社における乗務員ダイヤ作成システム

Crew Diagram System Based on Object-Oriented Method

柳 信治* Nobuharu Yanagi 山本勝則**** Katsunori Yamamoto
 岡崎敏郎** Toshirô Okazaki 梶井金徳**** Kanenori Kajii
 矢野純一*** Jun'ichi Yano 笠間誠一*** Seiichi Kasama



乗務系統ダイヤ(ダイヤグラム)を編集する画面例

上段が編集時の乗務系統ダイヤ表示部、下段が元データである車両系統ダイヤの山ダイヤ表示部となっている。乗務系統ダイヤとは、何時にどの駅からどの列車に乗り、どの駅でどの列車に乗り換えるといった、乗務員の行路を示す。

西日本鉄道株式会社では、電車総合情報システム構築の一環として、乗務員の運用ダイヤである乗務系統ダイヤの自動作成システムを開発した。

乗務系統ダイヤは、労働協約をはじめ、乗務員の所属する乗務所や交替駅の特性など多くの要素を考慮して作成する必要があるため、これまでは処理手順が明確ではなく自動化が困難であった。そこで、機能の修正や拡張が容易なオブジェクト指向技術を

用いて、段階的に完成度を高めていくアプローチを選択した。

初期システムに対し、約100項目の改善を段階的に反映した。いずれも、局所的で小規模な改造作業で実現することができた。オブジェクト指向特有の柔軟性と強力な整合性チェックにより、迅速な対応が可能となり、システムの実用化を早期に達成することができた。

* 西日本鉄道株式会社 電車局営業部 ** 西日本鉄道株式会社 システム計画室 *** 日立製作所 情報システム事業部
 **** 日立システムエンジニアリング株式会社

1 はじめに

西日本鉄道株式会社では、鉄道業務全般の自動化を目指す電車総合情報システム構築の一環として、乗務員ダイヤ作成システムの開発に取り組むことになった。これは、列車基本ダイヤに車両運用を決定した車両系統ダイヤから、乗務員の運用ダイヤである乗務系統ダイヤを自動スケジューリングするものである。

スケジューリングには、労働協約をはじめ、乗務員の所属する乗務所や交替駅の特性など、多くの要素を考慮する必要がある。これまで専門家がまったくの手作業で数か月を費やしていた割り付け業務は、定型的な処理手順がなく、問題の解法も不明りょうであった。そこで、試作品を何度も改修することにより、徐々に完成度を高め、そして実用レベルに到達する方法に最もふさわしいオブジェクト指向技術を適用することになった。

ここでは、乗務員ダイヤ作成システムの開発経験を踏まえて、オブジェクト指向適用の有用性について述べる。

2 システムの概要

乗務員ダイヤ作成システムの目的は、より少ない系統本数で乗務員の運用を実現することである。そのためには、むだな待ち時間をなくして実乗務時間比率が高い勤務の作成に主眼が置かれる。同時に、基準拘束時間を守り、休憩や付帯労働といった労働条件を詳細に取り決めてある労働協約を順守する必要がある。システムの概略を図1に示す。なおシステム環境は、日立のクリエイティブワークステーション3050RXと、HI-UX/WE2、X Window^{※1)}、日立Motif^{※2)}、C++などのソフトウェアで構成している。

3 オブジェクト指向技術選択の理由

乗務系統ダイヤの作成はスケジューリング問題である。組合せ条件は複雑で、定型的な処理手順はなく、問題の解法も不明りょうであった。専門家から知識の獲得を試みたが、部分的でかつ断片的なものでしかなかった。そこで、試作品を何度も改修し、徐々に完成度を高めて

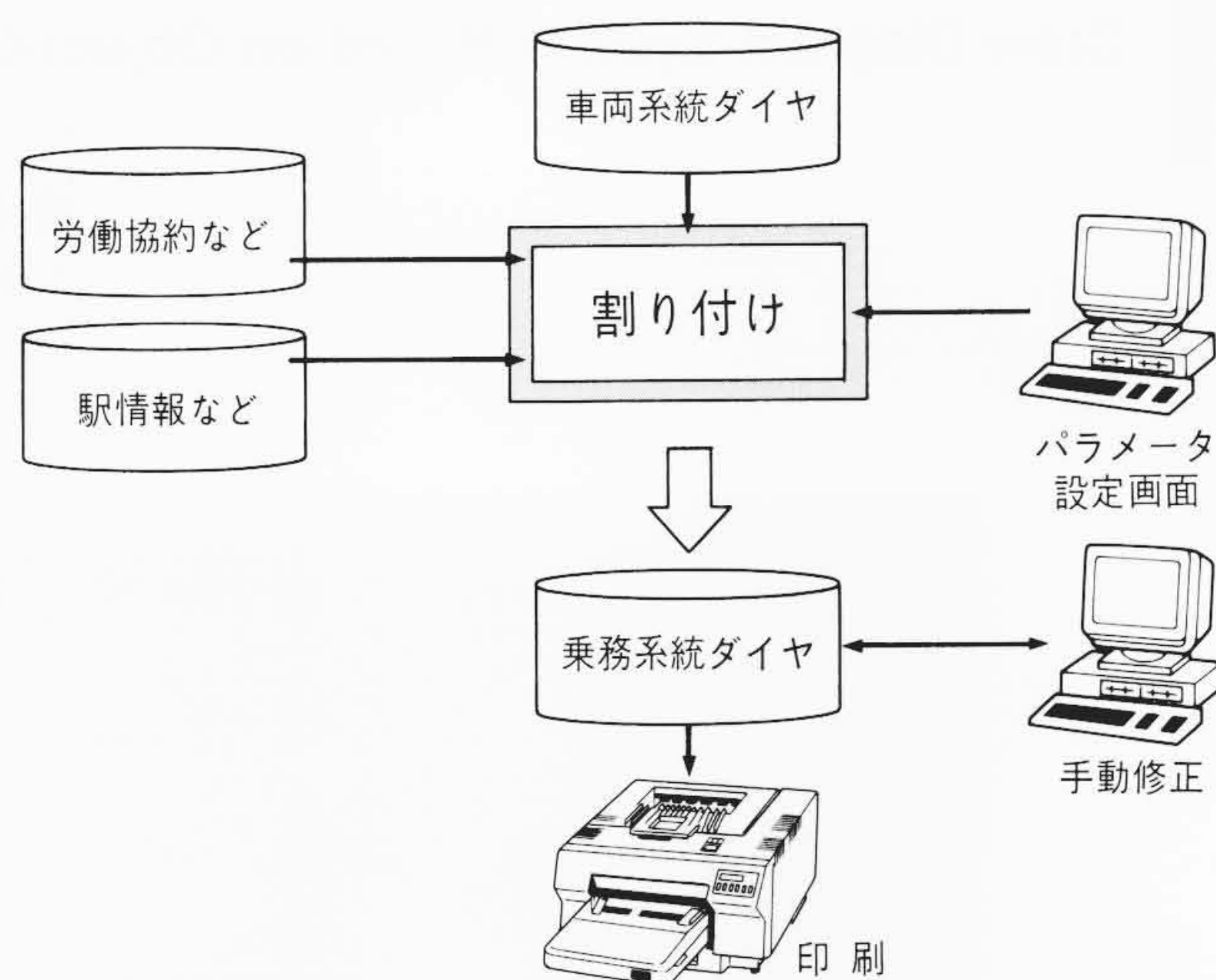


図1 乗務員ダイヤ作成システムの概要

元データとなる車両系統ダイヤを読み込み、労働協約や駅情報などの制約条件と、入力された戦略パラメータによって自動割り付けを行う。

いくというプロトタイピングアプローチ方式が最も現実的な方法であると判断した。

プロトタイピングアプローチ方式によるスケジューリング問題に対しては、これまでエキスパートシステム構築ツールを利用してきた。しかし、過去のツールの利用経験や今後の技術動向などを含めて調査、検討を行った結果、次に述べる理由からオブジェクト指向技術の適用を決定した。

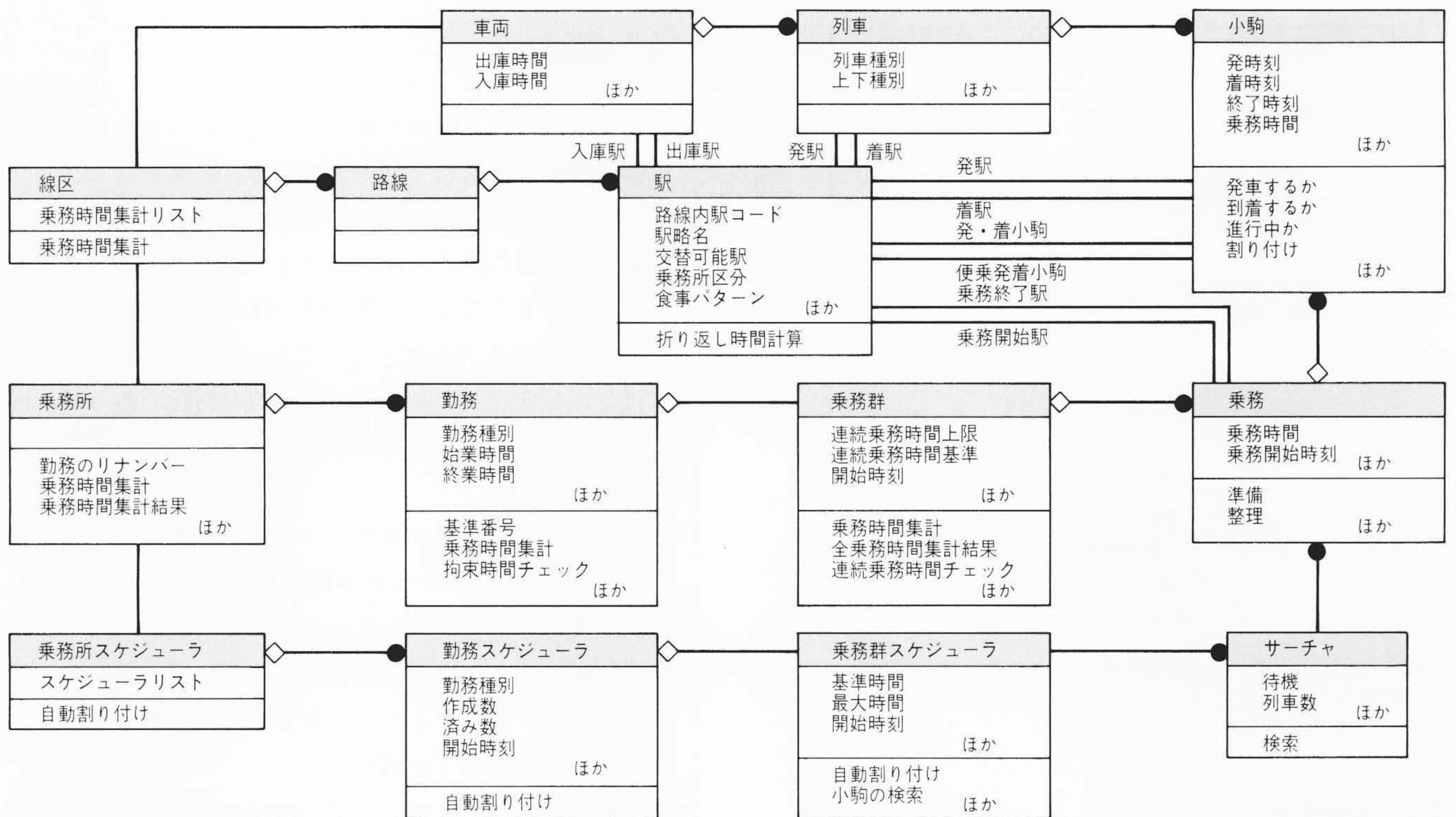
- (1) カプセル化によってモジュールの独立性の向上が図れ、改修に際し局所的な変更で対応できる。
- (2) 継承を用いた差分プログラミングにより、プログラムの作成、修正が容易になる。
- (3) 多義性により、処理の共通化、個別化が管理しやすくなる。
- (4) コンパイラの強力な最適化によって高速実行できる。
- (5) 流通しているクラスライブラリの利用が可能であり、ソートや抽出など汎(はん)用的なアルゴリズムのコーディングが不要になる。

4 クラス分析

OMT(Object Modeling Technique)により、システムに要求されるふるまいをベースにオブジェクトを抽出し、オブジェクトの果たすべき役割と責任の両面から分析した(図2参照)。動的モデルとして、状態図やイベントトレースを作成し、アルゴリズムを表現した。これによって処理の流れが明りょうになり、クラスに分担させるメソッドの仕様も理解しやすくなった。

※1) X Window Systemは、米国X Consortium, Inc.が開発したソフトウェアである。

※2) Motifは、Open Software Foundation, Inc.の商標である。



注：記号説明 ◇— (集約), ●— (複数), — (関連)

図2 OMTによるオブジェクト分析の例(一部)

ボックスがクラスを表し、上段がクラス名、中段が属性、下段がメソッドを示している。

ただし、OMTによる分析の際に集約関係を集合の包含関係の要領で表現したためC++言語と相性が悪く、言語実装の段階で大幅にクラス構造の見直しを行った。集約の表現を各クラスに共通な性質でまとめたものを上位の「抽象クラス」として定義し、階層的なクラス体系とした。「抽象クラス」の定義により、下位クラスでは個別の異なる部分だけ作成すればよいことから、差分プログラミングの効果が発揮できた。

5 改善作業

このシステムでは約100項目の改善要望を20回にわたりエンハンスした。ここでは、追加機能の一部について述べる。

5.1 バックトラックの実現

割り付け率向上のため、割り付け失敗時の組合せを学習するブラックリスト方式と呼ぶ機能を組み込んだが、それでも割り付け率は50%程度と低かった。そこで、サーチした小駒^{※)}候補リストを保管しておき、割り付け時

の小駒にチェーンを張り、割り付けのやり直し(バックトラック)を再帰的に実行できるようにした。小駒クラスに次候補リストという属性を追加し、取り出し関数を再帰呼び出しするメソッドを作成することで、一般的には難易度が高いといわれているバックトラックアルゴリズムを実現した。改造規模は約50ステップであった(図3参照)。

5.2 部分自動と部品活用

手動操作の際に、割り付け可能な系統を候補として提示してほしいという要望があった。これは、手動操作時に自動割り付けのアルゴリズムを起動して、割り付け可能な系統を探し出す部分自動処理であり、乗務系統を「仮に」作成するところが新しい問題であった。そこで、「仮勤務」という勤務の派生クラスを作り出すことで対応した。性格は勤務だが、勤務の仲間ではない位置づけで、その違いの部分だけを差分プログラミングした。さらに、仮勤務を常に別の場所に保管して、ある程度の塊として割り付けに利用できれば、より高度な組合せを実現できることがわかり、「中駒」と呼ぶ半完成品の部品として派生クラス化した。

※) 小駒とは、組合せの元データとなる乗務の最小単位のことである。

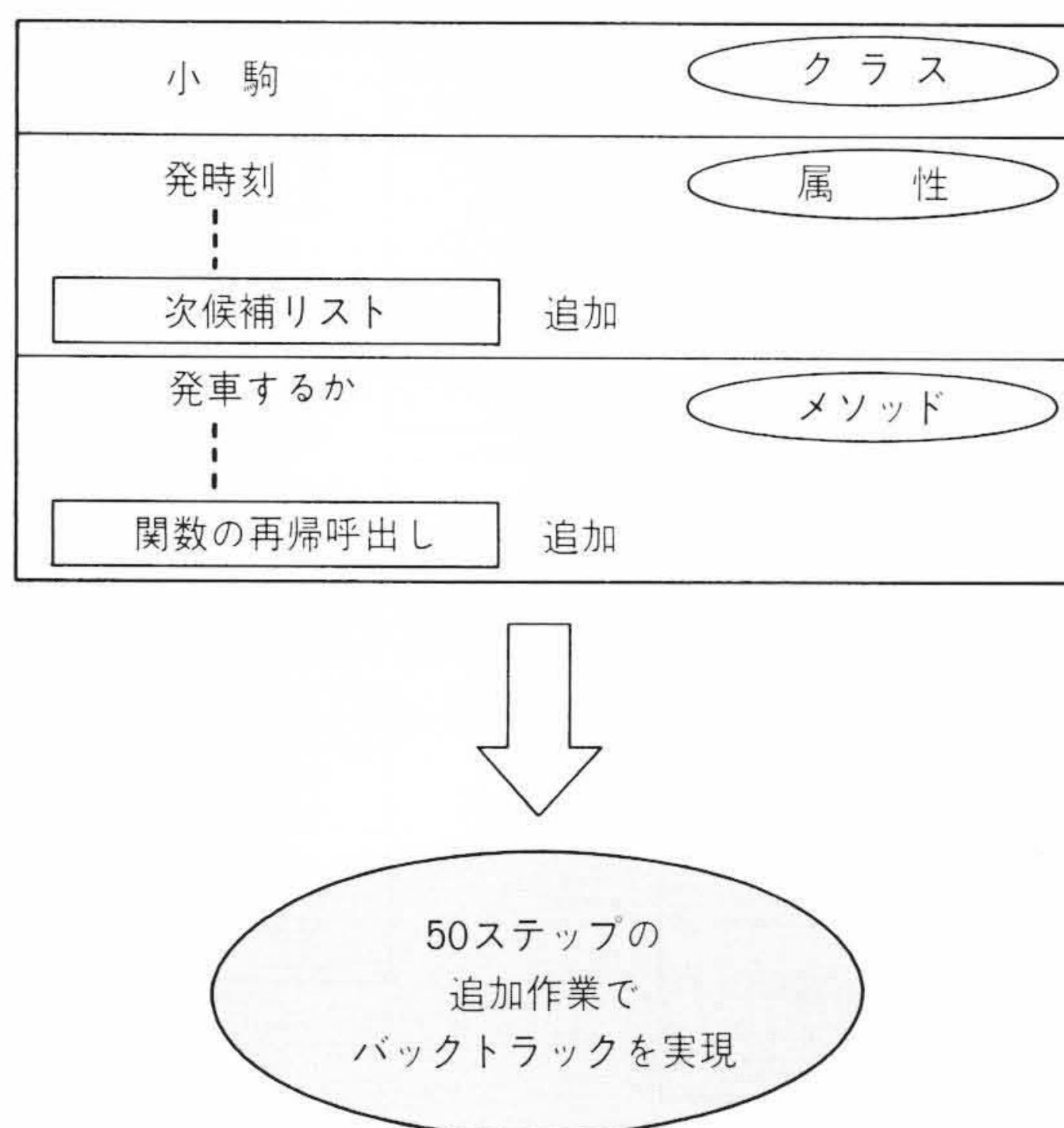


図3 バックトラック機能の追加イメージ

小駒クラスに、属性とメソッドを一つずつ追加しただけでバックトラック機能を実現した。

5.3 模倣作成

模倣作成とは、過去の乗務系統を手本にして、別の車両系統ダイヤに対し割り付けを試みる機能である。部分的なダイヤ改正の場合は、全面的に作り直す必要はないし、また全面的に変わってしまうと内外に説明がつかない。そこで、勤務情報から車両小駒を特定する「模倣作成」という処理の追加を行って対応したことにより、クラス全体の体系は変更することなく機能が実現できた。勤務情報の乗務時間に対し「模倣誤差時分」というパラメータを設け、設定値に従って比較許容誤差を考慮し、何度もトライできるようにしている。追加したメソッドは約100ステップで、作業日数も数日で終了した。

6 適用効果

- (1) オブジェクト指向を適用したスケジューリングクラス部の規模は、約11キロステップで、クラス数は24クラスであった。差分プログラミングの徹底によってステップ数が最小限に抑えられたことと、オブジェクトの独立性の向上の2点により、保守性の良いシステムとなっている。100項目もの改善を行っても、ステップ数についてはほとんど変化がなかった。経験的な判断であるが、従来の手法であれば、改善作業の結果、倍以上の規模になったと想定する。
- (2) 改善作業は、クラスやメソッドを一部修正すればよく、変更個所が少ないため短期間、少工数で対応できた。今回の改善作業にかぎっていえば、従来手法の約 $\frac{1}{3}$ の工数で実現できたと考えられる。
- (3) 汎用のクラスライブラリを効率的に使用することにより、生産性の向上が図れた。
- (4) 処理性能は、オブジェクト指向の適用によって当初の想定値を確保することができ、満足のいく結果が得られた。

7 おわりに

条件が複雑で仕様が不明りょうな問題は、これまでシステム化が困難であったが、オブジェクト指向技術を適用したプロトタイピングアプローチによって解決することができた。

現在は、GUI(Graphical User Interface)を用いたオブジェクト指向型の開発支援ツールも整備してきている。今後は、これらのツールを用いた開発により、いっそうの生産性の向上に努めていく考えである。

参考文献

- 1) J. ランボー：オブジェクト指向方法論OMT モデル化と設計、トッパン(1992-7)
- 2) 吉田：TURBO C++/BORLAND C++によるオブジェクト指向狂詩曲、技術評論社(1992-2)
- 3) 柴田：CプログラマのためのC++入門、ソフトバンク(1992-8)
- 4) スコット、外：Effective C++ C++の50の急所、ソフトバンク(1993-5)
- 5) 玉置：歴史でわかるオブジェクト指向開発の真価、日経コンピュータ、日経BP社(1993-9)
- 6) 日経コンピュータ：特集 ソフト生産性は3倍になる、日経BP、1994.6.27
- 7) K. Gorlen：NIH Class Library Reference Manual Revision 3.10, National Institutes of Health (1990-4)
- 8) 梶井、外：鉄道輸送業におけるオブジェクト指向による乗務員ダイヤシステムの開発、情報処理学会、第50回全国大会論文集(5) (1995-3)