
オブジェクト指向による開発生産性の向上

Increasing System Development Productivity by Applying Object Technology

木下 茂行(Shigeyuki Kinoshita) 佐能 克明(Katsuaki Sanou) 波元 秀世(Hideyo Namimoto)

要旨：

川崎製鉄ではシステム開発の生産性向上を狙って、オブジェクト指向の適用検討を進めている。オブジェクト指向では、利用者にとってわかりやすいオブジェクトを用いてモデル化を行い、オブジェクトを単位とする部品を再利用することでプログラムを作成する。また、大規模システムやクライアント／サーバといった新しいコンピュータ環境にも適した技術である。大規模システムのリフレッシュを対象に、オブジェクト指向技術を用い手効率的に開発するための技術を開発中である。これは、既存システムから再利用可能なオブジェクトを抽出する技術と、それを用いて効率的に分析・設計を実施する技術からなる。また、オブジェクト指向技術を実際に適用してシステム開発を行うことで、その有効性を確認した。

Synopsis：

Kawasaki Steel has been endeavoring to apply object technology in order to increase system development productivity. Object oriented systems are modeled by objects which are familiar with users, and built by re-using existing software components. It is applicable to large scale systems, or new computing environment such as the client/server one. Kawasaki Steel has now developing a methodology to migrate large-scale, mission critical legacy systems to object oriented systems. It consists of methods to extract re-usable software component from existing systems, and to analyze and design new systems by using them. Object technologies were proved to be effective by applying them to new system development.

(c)JFE Steel Corporation, 2003

本文は次のページから閲覧できます。

Increasing System Development Productivity by Applying Object Technology



木下 茂行
Shigeyuki Kinoshita
情報システム部 シス
テム技術室 主査(課
長)



佐能 克明
Katsuaki Sanou
情報システム部 シス
テム技術室 主査(課
長)



波元 秀世
Hideyo Namimoto
情報システム部 シス
テム技術室

要旨

川崎製鉄ではシステム開発の生産性向上を狙って、オブジェクト指向の適用検討を進めている。オブジェクト指向では、利用者にとってわかりやすいオブジェクトを用いてモデル化を行い、オブジェクトを単位とする部品を再利用することでプログラムを作成する。また、大規模システムやクライアント/サーバといった新しいコンピュータ環境にも適した技術である。大規模システムのリフレッシュを対象に、オブジェクト指向技術を用いて効率的に開発するための技術を開発中である。これは、既存システムから再利用可能なオブジェクトを抽出する技術と、それを用いて効率的に分析・設計を実施する技術からなる。また、オブジェクト指向技術を実際に適用してシステム開発を行うことで、その有効性を確認した。

Synopsis:

Kawasaki Steel has been endeavoring to apply object technology in order to increase system development productivity. Object oriented systems are modeled by objects which are familiar with users, and built by re-using existing software components. It is applicable to large scale systems, or new computing environment such as the client/server one. Kawasaki Steel has now developing a methodology to migrate large-scale, mission critical legacy systems to object oriented systems. It consists of methods to extract re-usable software component from existing systems, and to analyze and design new systems by using them. Object technologies were proved to be effective by applying them to new system development.

1 緒 言

当社は、日本における先進的なコンピュータ・ユーザとして、1950年代から情報システムの開発・利用を始め、60年代には他社に先駆けてオンラインシステムを稼働させている。それ以来多くのシステムの開発・利用を行っており、現在では全社の情報システムの総規模は、COBOLで1億ステップに迫ろうとしている。この間、システム開発・保守の生産性向上に向けて順次さまざまな対策が講じられており、開発生産性としては高い水準に到達している。しかしながら、近年だけを見ると、有効な技術が枯渇しつつあり、このままでは大規模システムの開発生産性を飛躍的に向上させることは困難な状況にある (Fig. 1)。

一方、バブル経済崩壊後の内外の急激な変化とそれに迅速に対応できる経営体制の確立、特に円高によるコスト競争力の喪失からの早期の回復が求められている。また、組織変更へのリアルタイムなシステム改造も必要とされる。情報システムとしてこれに応えるためには、開発・保守の生産性の飛躍的な向上などによるコストダウンと、迅速なシステム開発が必要である。このような経営からの要請にこたえるべく、状況を打破して飛躍的な生産性向上を達成することが可能な技術として、オブジェクト指向が今注目されている。

川崎製鉄でも、1993年から重点的にオブジェクト指向の適用検討に取り組んでおり、生産性向上に対して有効な技術であることを確認した。1994年度はさらに検討を拡大して、実際にオブジェクト指向を適用してシステム開発を行うとともに、大規模な基幹システムをオブジェクト指向に移行する (リエンジニアリングと呼んで

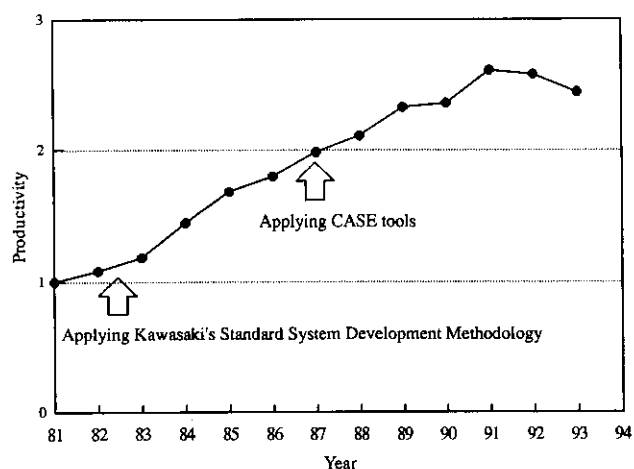


Fig. 1 Change in mean development productivity throughout company (representing relative values as 1981's productivity to 1)

* 平成7年4月5日原稿受付

いる)方法の検討も実施している。

本稿では、このようなオブジェクト指向を中心とする、システム開発に対する生産性向上活動を報告する。

2 オブジェクト指向技術について

2.1 オブジェクト指向の概要

オブジェクトとは、利用者にとって意味のある単位(スラブやコイルなどの物、圧延などの行為、オーダなどの業務上の振る舞い、画面や帳票などのシステムの情報、など)に、データとそれに関係した処理手続きをまとめたものである¹⁻³⁾。オブジェクト指向のモデルの例を Fig. 2 に示す。

オブジェクト指向では、システム分析、設計、プログラミング、データベースなど、システム開発のすべての局面で、Fig.2 に示すようなモデルを使用する。すなわち、業務分析のフェーズからプログラミングフェーズまで、基本的に同じ図式表現を使用することになる。このため、従来のシステム開発では要求仕様とプログラムの距離が離れていたのに対し、オブジェクト指向においては両者の距離が近いといえる。

オブジェクト指向は、ソフトウェアの産業革命であるともいわれている。これは、従来の一品ごとの手作りから、オブジェクトを単位とする標準部品を使った大量生産への移行を意味する。将来のソフトウェア技術の主流となることは多くの人が認めるところである。

2.2 オブジェクト指向への期待

オブジェクト指向は、前節で述べたような特徴を持つが、これが生産性向上に効果があると考えるのは、次のような点による。

2.2.1 モデルのわかりやすさ

従来のシステムでは、「データ」と、それをコンピュータで取り扱う「処理手続き」を個別にモデリングしている。一方、オブジェクト指向では、実体システムに出現する「物」をできるだけそのま

まモデリングする。このため、オブジェクト指向で開発されたシステムは、内容が実務部門やエンドユーザにとってわかりやすいものとなる。このわかりやすさは、実務部門とシステム部門のコミュニケーションの向上をもたらし、要求仕様の決定や効率化や、保守性の向上をもたらす。

2.2.2 RAD 実現の容易性⁴⁾

最近、従来に比べて飛躍的な短期間での開発を可能とする RAD (rapid application development) と呼ぶシステム開発方法論が注目されている。これを支える技術として、オブジェクト指向は重要な位置を占めている。

これは、前述の業務分析のフェーズからプログラミングフェーズまで基本的に同じ図式表現を使用できるという特徴から、工程の短縮、後戻りを容易にし、それによるプロトタイピングを可能にするからである。また、後述の部品化・再利用の容易性から、開発部分の圧縮が可能になる。

2.2.3 部品化・再利用性の高さ

オブジェクト指向が生産性の向上に最も寄りできるのは、この性質からである。この部品化・再利用性の高さは、オブジェクト指向のもつ次のような特徴によりもたらされる。

(1) 情報隠蔽

情報隠蔽とは、オブジェクトの内部のデータなどを外に見せないことである。そのオブジェクト外からは、メッセージ・パッシングという形態のみでオブジェクトにアクセスする。これにより、オブジェクト間の依存関係を少なくでき、高度なモジュール化が実現できる。この結果、オブジェクト単位のシステムの構築や変更が容易に行えるようになる。また、保守の局面においては、業務分析とプログラムの距離の近さから、保守すべき箇所の発見の容易さをもたらす。

(2) 継承機能

オブジェクト間で、概念の抽象—具体関係で表現するとき、例えば、「人間」というオブジェクトを抽象化したものは「哺乳類」である。これを、「人間」は (is_a) 「哺乳類」であるといい、抽象—具体関係を is_a 関係とも呼ぶ。このとき、抽

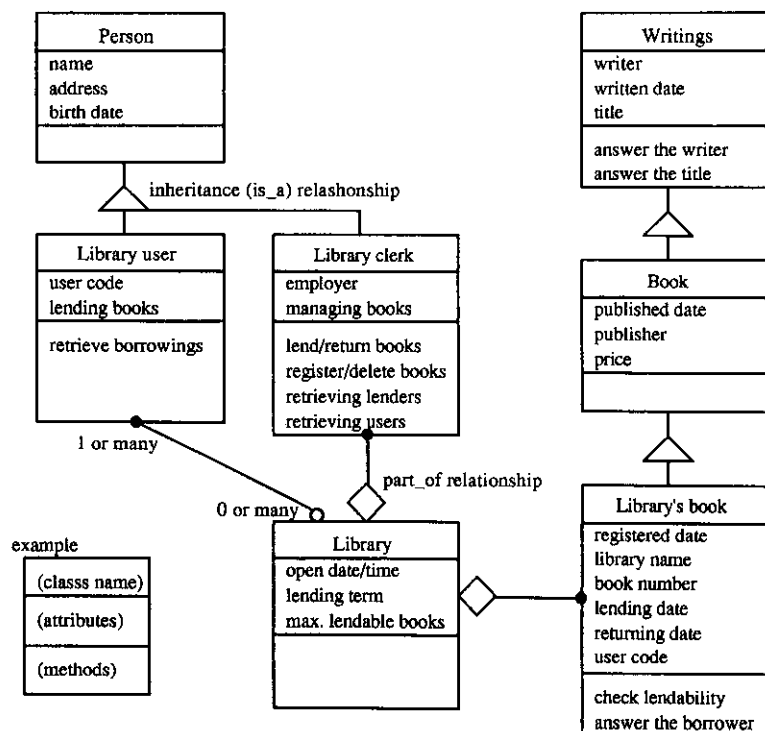


Fig. 2 An example of the object model (based on OMT notation)

象的な（上位の）オブジェクトの性質（データ、手続き）は、具体的な（下位の）オブジェクトも持つのが普通である。これを、性質の継承と呼び、下位のオブジェクトでは記述がなくても上位のデータ、手続きを持つ、すなわち上位のオブジェクトのデータや機能は、下位のものが再利用できる。下位のオブジェクトは上位のデータや機能と異なるもののみを記述すればよい（差分プログラミング）。機能をうまく抽象化すれば、継承機能による再利用はかなり有効となる。

(3) 多相性（ポリモフィズム）

上位のオブジェクトでの手続きの記述を行う場合、抽象的な記述が必要になる。この抽象的な記述を実現するものである。これは、複数のオブジェクト間で同じ手続き名（関数名、サブルーチン名）の記述を可能とし、どの手続きが実際に呼ばれるかを実行時に、ある変数にセットされたオブジェクト名などを見て決めるものである。これは、上位で記述できる機能（＝再利用可能なもの）を拡大する効果がある。

3 生産性向上に向けた取り組みの概要

3.1 従来からの取り組み

当社では、従来からデータ中心アプローチ（DOA）にもとづく、ウォーターフォール型のシステム開発手順の標準化、それをサポートする中下流 CASE ツールなどの適用を中心に、生産性向上に向けた取り組みを行っており、それなりの成果を得てきた。従来の取り組みを、以下で簡単に述べる。

3.1.1 システム開発手順の標準化

当社では、1981 年頃からシステム開発方法論として、システムックス社製の PRIDE を導入し、当社向けにカスタマイズを実施すると共に、全社共通のシステム開発手順として標準化し、適用している。PRIDE 導入当時は、アプリケーションとしては統合化、情報技術としてはデータベース化（DB/DC パッケージの使用）が盛んなときであり、一つのプロジェクトでの開発規模が 100 万ステップを超えるものが現れたときでもあった。そのような大規模システムの開発を、着実に、かつ効率よく行うことを目的としていた。

方法論としては、データ中心アプローチを採用しており、データ定義書を中心とするデータの記述と、システム→サブシステム→ロジック→プログラムと分解してゆく機能側の記述からなる。手順としては、記述を段階的に詳細化してゆく、いわゆるウォーターフォール型であり、ホスト中心の大規模システムの開発を対象としている。

これによる効果は、その後の大規模開発プロジェクトをスムーズに完了できたことや、生産性の伸びなどからも明らかである。また全社標準の開発手順として適用したことから、全社でのドキュメントの流通、開発手順・生産性指標の統一などの成果も得た。

3.1.2 CASE ツールの適用

上記の開発手順の標準化を受け、プログラム設計・プログラミング段階を支援するものとして、中下流の CASE ツール（本社はキャノンソフト製の CanoAid、製鉄所は富士通製の YPS/APG）を導入・適用している。これらは、データ中心アプローチと、プログラムのパターン化・部品化に基づくものである。開発環境としてはホスト端末以外にもワークステーションを使用するが、生成するプログラムはホストを対象とするものである。

これにより、プログラム設計・プログラミングのフェーズを中心

に、Fig.1 に示すように開発生産性の向上に寄与してきた。しかしながら、最近ではこれらによる新たな生産性向上は期待できなくなっている。また、生成したプログラムを直接変更した場合にツールの機能が使用できなくなるなどの、保守の局面での問題点も出てきている。

3.2 従来のアプローチの問題点

近年、ダウンサイジング、クライアント／サーバ・コンピューティングなど、メインフレーム中心とは異なる考え方が主流となっており、従来の開発手法・ツールが必ずしもこのような新しいコンピュータ環境にそぐわないものになっている。また、冒頭に述べたような経営環境の変化により、より安い開発コスト、短い開発期間を要求されている。これに対し、ウォーターフォール型の開発手順では、これに応えるのは難しくなっている。

3.3 新しい取り組みの概要

3.2 で述べたような従来の取り組みの問題点に対して、オブジェクト指向技術を中心とする新しい取り組みを開始している。オブジェクト指向技術を取り上げる理由は、次の二つである。第 1 の理由は、2 章で述べたようなシステム開発・保守の生産性向上に対して大いに貢献する可能性を持っていることである。現在のところ、大規模システム開発に対して、オブジェクト指向以外に大幅な生産性向上技術は見当たらない。第 2 の理由は、3.2 で述べたようなコンピューティング環境の変化に対して、オブジェクト指向技術・製品はメインフレームも含めた全体を対象とする技術を目指していることによる。ただし、現時点での製品はクライアント／サーバ環境を前提としたものが多い。

オブジェクト指向技術を適用したシステム開発形態の構想を Fig.3 に示す。オブジェクト指向の効果を最大限発揮して生産性を大幅に向上させるには、メインフレーム／サーバ／クライアントなどの機種に依存しない形でのソフトウェア部品の共有を図ることが重要である。このためのリポジトリを中心に考える。すなわち、再利用可能なソフトウェア部品を分析・設計レベルの文書情報と共にどこからでも利用できる形で保存しておき、さまざまなシステムの開発において、これらの部品を再利用することで生産性を向上させるとともに、開発期間も短縮する。

冒頭に述べたように現状では膨大なソフトウェア資産を保持しており、その資産を活用するという意味でも、ソフトウェア部品については現行の COBOL プログラムからある程度自動的に抽出することを目指す。すなわち、現行の COBOL プログラムから、データとそれに関連する処理（制約条件のチェック、データの導出）をひとまとめにカプセル化し、オブジェクト指向の部品とする。これにより、今後の主流となるリフレッシュ型のシステム開発においては、既存部品を再利用して少ない負荷で開発を行うことが可能となる。

基幹系のシステム開発においては、段階を踏んだ着実な設計がある程度は必要となる。これに対して、オブジェクト指向分析・設計を適用する。これは、上記の段階的な設計と、オブジェクト指向の特徴を活かした RAD 型のプロトタイプアプローチ（分析／設計／実装のサイクルを短い周期で繰り返すことにより、ユーザの要件を漏れなく反映したシステムを作成する）を融合したものである。分析・設計の段階では、先に述べた既存システムから抽出した部品や、他のシステムの部品などを再利用する。これをより広い範囲で実施するために、オブジェクトの設定単位の統一を行う。このために、オブジェクト分析方法の標準化や、オブジェクトの枠組みの設定な

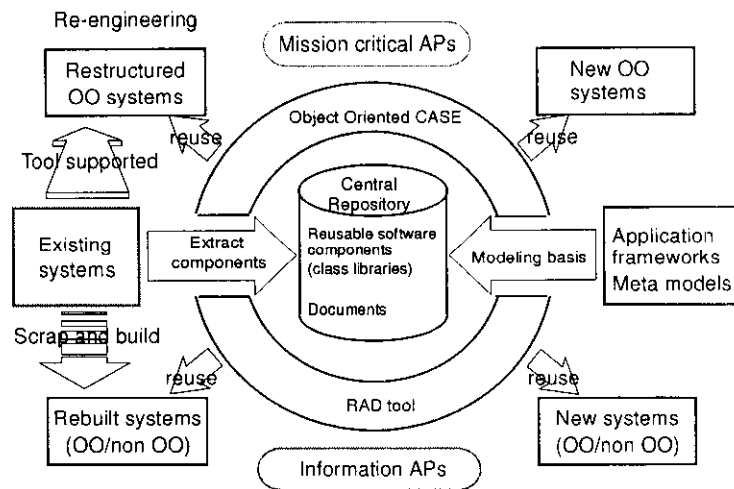


Fig. 3 Basic ideas for applying object technology to increase system development/maintenance productivity

どを行う。実装においては、開発者になるべくオブジェクト指向を意識させないように、CASE ツールなどによる自動化を行う。この理由は、一般に負荷がかかるといわれている既存の SE へのオブジェクト指向技術者としての再教育の負荷を軽減させるためである。

情報系のシステムの開発においては、開発期間の大幅な短縮とコスト軽減を狙った、徹底した RAD アプローチを採用する。これは、既存システム、他システムなどの部品を再利用するとともに、比較的簡単なシステムを効率よく作成するツール（ビジュアル・プログラミング・ツール、PC のデータベース・ツールなど）を使用することで可能とする。

4 オブジェクト指向適用に向けた活動の概要

現在、当社では、前章で述べた考え方にしたがって生産性向上に向けた活動を実施している。ここでは、そのうちの基幹システムを対象としたオブジェクト指向技術の適用に向けた活動の内容について簡単に述べる。

4.1 基幹システムのリバースエンジニアリングの検討

当社においては、情報システム化は一通り終了しており、最近の大規模開発のほとんどは以前開発したシステムのリフレッシュとなっている。新旧のシステムの間では、機能的には差はあっても、データ（およびそれに付随する機能）は新規機能のために追加させるものを除いて、そのまま使われることが多い。従来のリフレッシュ方法では、旧システムの資産は、ドキュメントは参照されることがあっても、プログラムそのものはほとんどすべてが再作成となっている。このようなシステム資産を、設計情報、あるいはプログラム部品として再利用することにより、開発生産性の向上に大いに寄与できるはずである。

この検討では、既存のホストの COBOL プログラムから、データを中心とする再利用可能な部品（プログラム部品、設計情報）を抽出することを目的とする。現在、リフレッシュの候補として上げられている古いシステムの一部（約 5 万ステップ）を対象に、検討を進めている。

リバースの考え方は、データを中心に、データ中心アプローチの考えに基づいてその制約条件と導出操作を COBOL プログラム中から抽出し、一塊のカプセルとするものである⁵⁻⁷⁾。リバースのプ

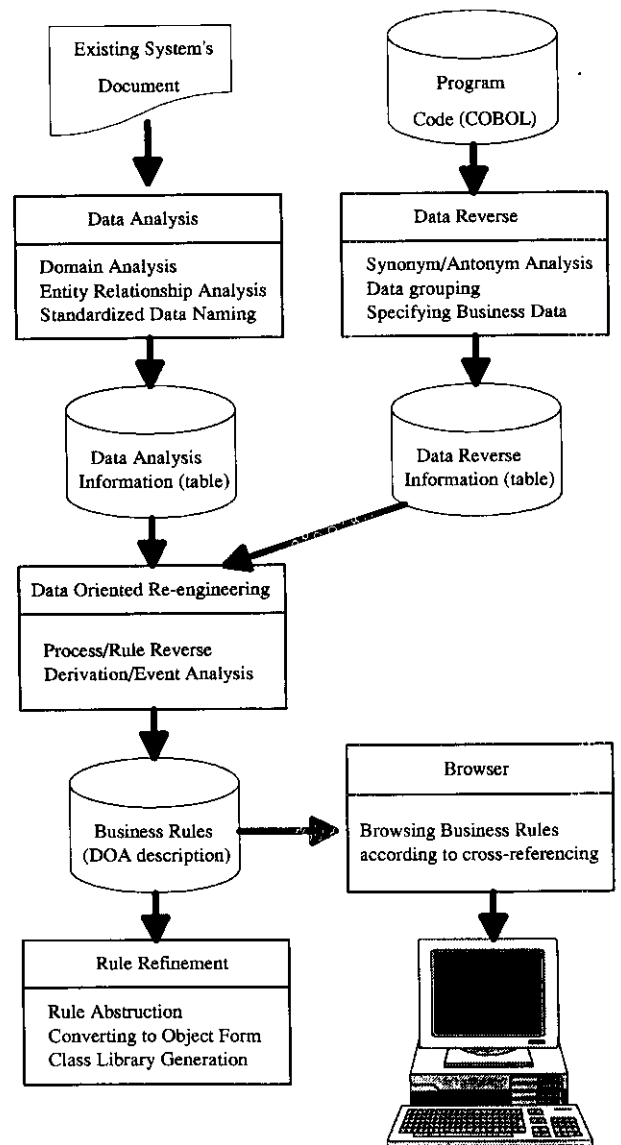


Fig. 4 Reverse engineering process

ロセスを Fig. 4 に示す。現在、この考え方およびツールにしたがって、従来の開発における「データ定義書」の記述レベルに相当す

る部品（業務ルールと称する）を抽出できることを確認している。

4.2 オブジェクト指向による業務分析法の開発

現在オブジェクト指向分析手法として、OMT 法⁸⁾、Booch 法や、IE (information engineering) に基づくもの (Martin/Odell 法⁹⁾、Ptech 法) などが提案されている。しかし、これらの手法は、いずれも既存システムが存在することを仮定していない。当社では、前述のように新規の大規模開発を対象とする場合、ほとんどが

既存システムを前提としたリフレッシュである。そこで、既存システムから得られる情報と、トップダウンの業務分析から得られる情報を融合した、基幹システムのリエンジニアリング向けのオブジェクト分析手法を開発した。

当手法の概要を、Fig. 5 に示す。当手法は、現在日本で普及している OMT などの手法の上流に位置付けられるものであり、次のような特徴を持つ。

- (1) トップダウンとボトムアップの分析を融合した手法である。
- (2) トップダウン分析では、責任主導、Use Case などの最新のオブジェクト指向分析の考え方を取り入れている。
- (3) ボトムアップ分析では、リバース・プロセスで得られるデータのライフサイクル情報に基づき、オブジェクトの候補を抽出する。

現在、リフレッシュ候補となっているシステム（前述のシステム全体、約 100 ワステップ）を対象に、この手法を適用してオブジェクト分析を実施中である。

4.3 リエンジニアリングを対象とする CASE ツールの検討

4.1 および 4.2 で述べた検討分野は、Fig. 6 に示すように、オブジェクト指向開発の最上流部分のオブジェクト指向分析 (OOA: object oriented analysis) の前半部分と、リバースエンジニアリングの部分である。基幹システムのリエンジニアリングを実施するには、4.2 で述べた分析手法の継続作業として、4.1 で述べた部品を使って分析・設計を進め、クライアント/サーバ・システムとして実装する必要がある。これを効率よく行うためには、CASE ツールを始めとする各種のツールを使う必要がある。また、リバースエンジニアリングの結果をオブジェクト指向分析・設計と融合させるための方法論の確立も必要である。

このため、最新のオブジェクト指向 CASE ツールを中心に、調査を実施している。残念ながら、現時点では当社の要求を完全に満たすツール（の組み合わせ）を見出すに至っていない。この分野のツール類は、技術的な進歩が早く、新しく良いものがつぎつぎに現れる。したがって、今後も継続して調査を行い、当社として最も良いものを選択する。

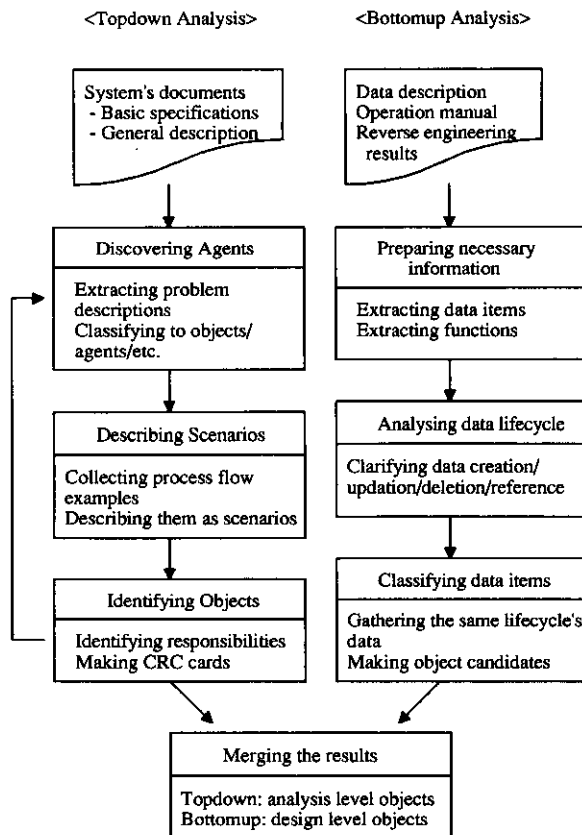


Fig. 5 Flow of the re-engineering oriented object analysis method

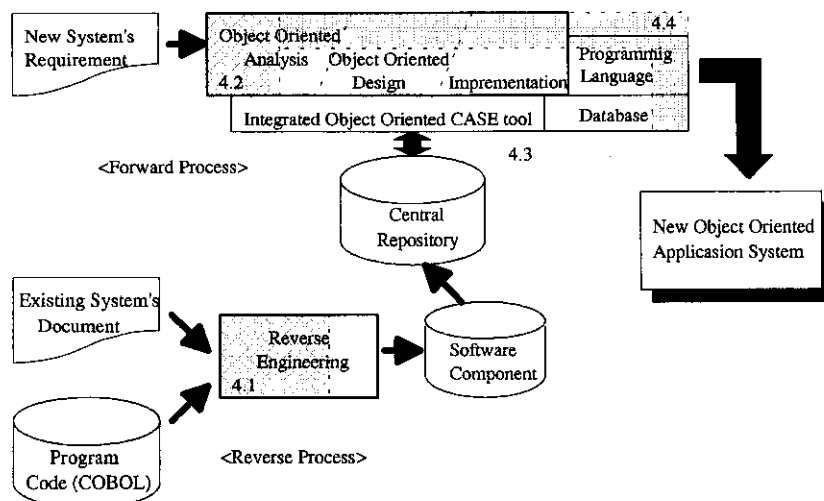


Fig. 6 Object oriented re-engineering process and region of the Kawasaki Steel's investigations (numbers such as 4.1 denote the paragraph)

4.4 プロトタイプ・システムの開発

オブジェクト指向開発の実践として、生産管理のプロトタイプ・システム (COBOL 換算 5 万ステップ規模) を言語/開発ツールとしてパークプレース社製の VisualWorks (Smalltalk), オブジェクト指向データベースとしてセルビオ社製の GemStone を採用して開発した。このシステムは、単にプロトタイプにとどまらず、これを現場に持ち込んで評価のための運用を実施中である。

本システムの開発において、生産性の評価を実施し、概略従来の COBOL ベースの開発に比較して 2 倍弱の生産性の向上と、オブジェクト指向の機能による規模の圧縮ができることを確認した。また、これは使用したツールの問題ではあるが、複雑な帳票出力処理は苦手であることも確認した。

5 今後の課題

Fig. 2 で示したようなオブジェクト指向の適用形態を目指して現在活動中であるが、その検討はまだ緒に着いたところであり、多くの課題が残されている。ここでは、そのうちの主なものを紹介する。

(1) オブジェクト技術者の育成

オブジェクト指向は、従来のシステム開発の方法とは異なった考え方・技術を使用するため、その技術者の育成には時間がかかる。また、技術として体系だった理論的なものではなく、多くの経験則の集大成されたものという色彩が強い。このため、育成の方法として OJT に頼らざるをえず、これが時間がかかる大きな原因である。

育成のためには、半年程度で開発が終了するような中規模のシステム開発にどんどんオブジェクト指向を適用することが必要である。これにより、経験を積んだ技術者を育成するとともに、開発手法の標準化や、使用するツールの評価などを蓄積し、オブジェクト指向の本格的普及の基盤としてゆく必要がある。

(2) 再利用技術の確立

オブジェクト指向の適用による生産性向上の効果の多くの部

分は、再利用に依存している。現在、再利用技術がある程度確立しているのは、画面 (GUI) 部品、データベースや通信などの外部とのアクセス部品などである。アプリケーション部品 (いわゆる、ビジネスオブジェクト) の再利用については全く確立されていない。また、大規模なシステムを対象としたような再利用部品の管理技術 (リポジトリなど) についても技術の確立が遅れている。

このうち、当社として検討すべき課題は、アプリケーション部品の再利用技術であろう。これは、当社の基幹システムにおけるビジネスオブジェクトの分類、それぞれの内容の明確化などを行うことで、再利用方法が明確になってくると考える。ビジネス・オブジェクト¹⁰⁾については、アメリカを中心に議論が始まっており、これらの成果も取り入れたい。

6 結 言

当社におけるオブジェクト指向の適用検討の状況について述べた。まとめると、次のとおりである。

- (1) オブジェクト指向は、オブジェクトを単位とする標準部品を使ってソフトウェアの大量生産を可能にするもので、モデルのわかりやすさ、RAD (rapid application development) 実現の容易性、および部品化・再利用性の高さにより、生産性を飛躍的に向上させることができる。
- (2) また、対象となるコンピューティング環境が広く、大規模システムの開発・保守に効果的である。
- (3) 基幹システムのリフレッシュにおけるオブジェクト指向の適用では、既存システムから再利用可能な部品を抽出が可能なこと、それを用いて効率的な開発ができることが確認できた。
- (4) 生産管理プロトタイプの開発では、従来より 2 倍弱の生産性の向上と規模の圧縮をはかることができた。

しかしながら検討すべき課題はまだ多く、今後も引き続き大規模システムへの適用も含めた検討を実施して行く。これからの検討は必ずしも平坦な道ではないが、情報システム化が今後の経営の大きな位置を占めるといわれる中で、成果は大いに貢献できるものと信じる。

参 考 文 献

- 1) 中谷多哉子: 「オブジェクト指向源流分析から設計へ」, (社) システム総合研究所第 104 回情報・通信分科会, (1992), 56
- 2) 春木良且: 「オブジェクト指向への招待」, (1989), 191, [啓学出版]
- 3) 青木 淳: 「オブジェクト指向システム分析入門」, (1993), 218, [ソフトウェア・リサーチ・センター]
- 4) ジェームス・マーチン: 「ラビッド・アプリケーション・デベロップメント I, II」(1994), 283, 269, [リックテレコム]
- 5) 堀内 一, 飯田啓三: 「データ中心によるリエンジニアリングの方法」, 情報処理学会情報システム研究会, 情報システム 42-4, (1993), 37-46
- 6) 山川敦夫, 宮本信夫, 上林高治, 石本真希, 秋庭真一, 山村伸幸, 堀内 一: 「データ中心分析によるプログラム論理の抽出」, 情報処理学会情報システム研究会, 情報システム 42-11, (1993), 107-115
- 7) 山川敦夫, 上林高治, 石本真希, 宮本信夫, 飯田啓三, 堀内 一, 秋庭真一, 山村伸幸: 「データ中心アプローチによるリエンジニアリング支援技術に適用実験」, 日立論評, 75 (1993) 11, 40-44
- 8) A. ランボー, M. プラハ, W. プレメラニ, F. エディ, W. ローレンセン: 「オブジェクト指向方法論 OMT」, (1992), 544, [トッパン]
- 9) J. Martin, J. J. Odell: "Object-oriented Analysis and Design," (1992), 513, [Prentice-hall]
- 10) D. A. Taylor: "Business Engineering with Object Technology," (1995), 188, [John Wiley & Sons]