

TR-0838

産業界におけるソフトウェア・
エンジニアリングの動向

本位田 真一 (東芝)

April, 1993

© 1993, ICOT

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03)3456-3191 ~ 5

Institute for New Generation Computer Technology

標題：産業界におけるソフトウェア：エンジニアリング
の動向

著者名：本位田真一

所属機関：（株）東芝 研究開発センター
システム・ソフトウェア生産技術研究所

所在地：〒210 川崎市幸区柳町70

英文標題：Software Engineering in Industry

英文著者名：Shinichi Honiden

英文所属機関名：Toshiba Corp., R&D Center,
Systems & Software Engineering Laboratory

標題：産業界におけるソフトウェア・エンジニアリング
の動向

著者名：本位田真一

所属機関：（株）東芝 研究開発センター
システム・ソフトウェア生産技術研究所

所在地：〒210 川崎市幸区柳町70

英文標題：Software Engineering in Industry

英文著者名：Shinichi Honiden

英文所属機関名：Toshiba Corp., R&D Center,
Systems & Software Engineering Laboratory

連絡先：〒210 川崎市幸区柳町70

（S研） 本位田真一

Tel. 044-548-5474

Fax 044-533-3593

本文原稿枚数

実枚数：16枚 => 換算枚数：32枚

図：2点

署名：

1. はじめに

産業界においてソフトウェア危機が叫ばれて以来久しい。その間、構造化プログラミングなどの技術がソフトウェア・エンジニアリングの成果としてソフトウェア開発の下流工程を中心に導入され、ソフトウェアの生産性の向上に寄与してきた。しかしながら、ソフトウェアの生産性を飛躍的に向上させるのは、上流工程の充実に他ならない。そのために、1970年代後半から、多くの研究がなされ、実務ソフトウェアへの適用評価が繰り返されてきた。筆者も約15年間に渡り、上流工程における様々な手法、ツール、方法論の研究開発に従事してきた。本稿では、筆者らの最近のアプローチを事例として紹介することによって、産業界において、ソフトウェア・エンジニアリングが上流工程に対してどの程度適用され、何が課題なのかを明らかにすることを試みる。なお、紙面の都合上、それぞれのアプローチに関しては参考文献をあげるにとどめ、それらの解説は不十分であることをお許し願いたい。

2. 上流工程の支援手法

上流工程を支援する手法に対する要件としては以下の項目が考えられる。

- ・ ユーザとシステム開発者との間のコミュニケーションの道具となり得ること
- ・ 仕様の正しさの検証・確認が容易にできること。
ここで、仕様の正しさとは、ユーザの意図を反映しているかどうか、および仕様として矛盾がないかどうかの両方を指す。
- ・ 下流工程への連結が自然であること。上流工程での作業プロダクトが下流工程に対して有効な情報となり得

ること。

- ・対象世界の自然なモデリングが可能であること。
- ・モデル化された仕様の記述性、理解性が十分に高いこと。
- ・仕様の再利用性，変更容易性が高いこと。
- ・実務での活用の観点では、さらに、習得のしやすさも重要な項目である。

上記の項目を満たすべく、従来より多くの研究がなされてきた。それらを思いつくままに列挙するならば、

- a. プロトタイピング
 - b. 操作的仕様記述
 - c. 知識ベース
 - d. 形式的仕様記述
 - e. 構造化分析・設計
 - f. オブジェクト指向分析・設計
 - g. 第4世代言語
 - h. CASEツール
 - i. 自動プログラミング
 - j. 制約指向
 - k. ネット仕様記述
 - l. 問題向き言語 (POL)
- などがある。

これらの研究の中には、第4世代言語、構造化分析・設計、プロトタイピングなど実務に寄与している手法も少なくないが、実用化という観点では多くの手法が研究段階であるといえる。これらの中で最近、特に注目されているのは以下の3つである。

- 1) 形式的仕様記述
- 2) オブジェクト指向分析・設計
- 3) CASEツール

そこで、本稿では、これらの3つの項目に関する筆者ら

のアプローチを中心に述べることによって、これらの技術の成熟度、実用可能性を探ることにする。

3. 形式的仕様記述手法

3.1 形式的仕様記述手法の課題

種々のソフトウェア仕様化技術のなかで、形式的手法は、それによって得られるソフトウェアの品質や信頼性を高める手法として広く認められている。しかし、リアルタイムシステムを対象とした場合、産業界では十分に活用されているとは言い難いのが現状である。形式的仕様が実用に供されるためには、少なくとも次の要件に対して何らかの形で応ずる必要がある。

まず、所望のシステムの性質をできる限り多く記述するための形式的手法を確立することである。これは、実務ソフトウェアに適用するに当たって最も要求される項目である。

第二としては、形式的手法を用いてどのように概略仕様からより詳細な仕様に展開するかといった、仕様展開プロセスを確立することである。形式的な記述の習熟するにはかなりの訓練と時間を必要とするため、仕様展開プロセスが確立していることが望ましい。また、これによって、複数の人に同一の概略仕様を与えたとき、得られるプロダクトにおいて、できる限り差異が生じないようにすることが可能となる。

第三には、形式的手法での記述内容を簡便な手法で検証・確認する手段が必要である。記述された仕様のバグを容易に見つけることのできる手段も、合わせて必要である。ここで、第二と第三の要件は、実務ソフトウェアにおいて、形式的仕様に経験の少ない人にとっては特に必要となる。

第四には、対象とする実務ソフトウェアの規模が大き

い場合には、全体を見渡すことのできる手段も必要である。

第五に形式的手法で記述された仕様から、プログラムを生成できることが要求される。いかに厳密に仕様を記述したとしても、それがプログラムに直接結びつかないのであるならば、実務現場での受け入れ可能性は極めて低くなる。

これらの要求項目に応ずるために従来より多くの仕事がある。たとえば第一の要求を満たすために、複数の形式的仕様を効果的に組み合わせる数多くの手法の提案がある。これは、単独の形式的手法によって所望のシステムの性質をすべて記述することは、現実的に困難であることによる。しかしながら、これらの手法がそのほかの要求項目を満たしているとは言い難い。そこで、筆者らはこれらの要求項目を満たすことを目指したシステム(MENDELS ZONEと呼ぶ)を提案したので紹介する[本位田92a]。

3.2 MENDELS ZONEによるプラント制御システムの記述

MENDELS ZONEは並列プログラムのためのソフトウェア開発環境である。特にICOTにおいて開発された並列論理型言語を対象としている。

MENDELS ZONEは上記の要求項目に対して以下のように解決策を提案している。まず、第一に複数の形式的手法として、代数的仕様と時制論理の組み合わせを採用する。代数的仕様によって静的な側面である機能やデータに関する性質を記述し、時制論理によって動的な側面である並行性、同期、排他に関する性質を記述するものである。第二に、リアルタイムシステムにおいて広く利用されているリアルタイムSA(Structured Analysis)を採用し、仕様展開プロセスを規定する。リアルタイムSAにおいては、DFD(Data-Flow Diagram)の記述に代数的仕様を適用

し、制御スベックの記述に時制論理を適用する。第三の課題に対しては、代数的仕様、時制論理には様々な体系が存在するが、できる限り自動検証可能な範囲に限定させる。第四の課題に対しては、DFDでの階層関係に従って、上位のDFDから下位のDFDへ、またはその逆というように自由に記述世界をtraverseできるようにし、全体を見渡すことを可能とする。第五の課題に対しては、オブジェクト指向設計の概念を導入する。

また、MENDELS ZONEにおける開発の流れは以下の通りである。

- (1) 所望のシステムに関するDFDを展開し、オブジェクトおよびそれらも構成するメソッドを抽出する。
- (2) オブジェクトのメソッドを関数としてとらえ、等式論理に基づく代数的仕様で記述する。
- (3) オブジェクトの内部構造をペトリネット表現で記述する。
- (4) メソッド間の実行順序に制約がある場合、そのタイミング制約を時制論理に基づく仕様記述言語で記述する。MENDELS ZONEは、その制約を満たすようにペトリネットの自動調整を行う。
- (5) オブジェクトをペトリネットエディタで結合する。
- (6) オブジェクト間の実行順序に制約がある場合も、メソッド間の場合と同様に記述、調整を行う。
- (7) 完成したMENDELプログラムを並列論理型プログラムにコンパイルする。

MENDELS ZONEの有用性を評価するために、実務ソフトウェアとしてプラント制御システムをその上で開発した[浦岡92]。このシステムは現実のプラント制御知識に基づいて動作するエキスパートシステムである。このシステムはすでに別のグループによって直接、並列論理型言語を用いて開発されたものである。MENDELS ZONEによる

形式的仕様の記述量としては約3700行であり、生成された並列論理型プログラムは約6200行であった。その結果、並列論理型言語による直接開発では、設計から完成まで、約10人月要したが、MENDELS ZONEを用いて開発した場合には、約5人月でシステムを完成させた。これは、前者で6人月要していたデバッグ時間を0.5人月に短縮できたことによる。オブジェクトのメソッド単位で検証を行いその正しさを確認したことが、結合してからのデバッグ時間を大幅に短縮させた。

3.3 考察

形式的仕様に関しては、従来より多くの研究が継続されているが、少なくとも国内では実用に供した事例は極めて少ない。これは、直接、形式的仕様での記述が現場にはなじまないことによる。MENDELS ZONEにおいては可能な限り、図形仕様を導入することによって緩和を図っているが、筆者らの開発経験を踏まえると、実務適用の観点では十分とは言えない。その改善のためには、検証系の洗練化とともに、方法論の定式化も今後の課題である。今後、多様化するソフトウェア・システムにおいて、ますます形式的仕様の必要性が高まることは必須である。また、形式仕様の規格化、標準化の動きも活発化する気配を見せている。地道な基礎研究と同時に、実用化研究も併せて望まれる。

4. オブジェクト指向分析・設計

4.1 オブジェクト指向分析・設計の課題

オブジェクト指向分析・設計は1990年代の方法論として注目されている。しかし、1970年代より、数多くの方法論が提案され、実務では充分には適用されずに消えていった。オブジェクト指向分析・設計も一時的な流行な

のであろうか？少なくとも他の方法論との相違点としては、下流工程においてオブジェクト指向言語がプログラミング言語として主流になりつつあることがあげられる。その結果として、上流工程から同一のモデル化手法を適用したいと思うのは当然のことである。しかしながら、上流工程においてもオブジェクト指向モデルが実務上有用であるかは、種々の適用事例を踏まえてみないと正確な判断はできない。筆者らは種々の適用体験を踏まえていくつかの知見を得た。

- (1) 教科書の知識だけからは現実の問題には対処しきれない。
- (2) 種々の方法論が提案され、それぞれ細部では異なっているが、いずれも本質的には同じである。
- (3) 仕様化プロセスはラウンドトリップ型の形態であり、現場に適用して行くためには、より詳細な仕様化プロセスを定義する必要がある。

また、現場からの要請としては次のような項目がある。

- (a) クラスライブラリの作成は極めて難しいのではないか。
- (b) オブジェクトの切り出しは個人差が生じるのではないか。
- (c) オブジェクトの有する状態の扱いが再利用性を低下させているのではないか。
- (d) 構造化分析・設計と何が異なるのか。

こうした現場の声を踏まえて、筆者らは具体的な実務の問題に様々な方法論を適用し、問題点を検出しそれらに対する対策を検討すべく作業を進めている。以下ではそれらのいくつかを紹介する

4.2 オブジェクト指向分析・設計への取り組み

4.2.1 構造化分析・設計との相違

オブジェクト指向分析手法と構造化分析手法の一つであるリアルタイムSAを画像ファイリングシステムの検索機能の分析に適用した[山城92]。オブジェクト指向パラダイムと構造化パラダイムの違い、およびその違いが分析作業の容易性と分析結果の理解性に与える影響について検討した。すなわち、それぞれの手法の共通点と相違点、分析・設計の役割分担、手順の実施容易性、表現の一意性、トレーサビリティの4つの観点から両手法の利点・欠点の検討を行った結果、構造化分析手法に対してオブジェクト指向分析手法が優位であった。

また、変更容易性については、変更が容易であるためには、まず仕様変更箇所の特定制が容易であり、次に変更作業そのものが容易であることが望ましい。両手法の評価結果より考察すると、仕様変更箇所の特定制は、それがシステムの環境や資源に関する変更であるならばオブジェクト指向分析、またシステムレベルの機能に関するものであるならばリアルタイムSAの方が特定し易いと考えられる。これは、どのような観点でモデル化しているかに依存するものである。また、仕様書の修正量としてみた場合の変更作業そのものは、変更の種類によらずオブジェクト指向分析の方が少なかった。

4.2.2 クラスライブラリの構築

オブジェクト指向分析において、何が難しいかという問いに対して多くの人がオブジェクトの切り出しの難しさ、クラスライブラリの構築の難しさをあげる。これは正しい指摘だと思う。しかし、はたしてソフトウェア開発者全員がオブジェクトを切り出し、さらにクラスライブラリを構築する必要があるのであろうか？これは必要ないを考える。クラスライブラリは、標準化された部品体系であるといえる。現場では従来から部門毎に地道に

標準化部品の構築を行なっている。その部門の中で最も信頼も厚く経験豊富な技術者（ここではスーパーS.E.と呼ぼう）が標準化部品を構築するのが望ましいが、クラスライブラリについても同様である。一方、一般のS.E.は、信頼のおけるスーパーS.E.の構築したクラスライブラリを修正しながら再利用していく。ここで、一般のS.E.のために再利用プロセスをきちんと定式化しておく必要はある。このような役割分担によって、従来からの労働集約型のソフトウェア開発体制を改めることもできると考える。

4.2.3 状態の扱い

オブジェクトは状態を有している。従来の部品が機能中心の静的なモジュールであるのに対して、オブジェクトは動的なモジュールであるといえる。これがオブジェクトの再利用の大きな弊害になっている。筆者らは、複数視点による状態の扱いによってこの問題の解決を試みている[入江92]。例として、総合情報システムをあげ、運用に関わる複数のオペレータの視点に注目した。仕様・運用変更はオペレータの関わり方の変更として捉える。もっと一般的に、複数のオブジェクトのある該当オブジェクトへの関わり方と位置づけても良い。総合情報システムにおいて、各オペレータからの視点で状態をモデル化する。このモデル化は変更されることはない。変更される箇所は、オペレータ間の順序関係である。そうした部分をオブジェクト内に状態として明記（具体的には並列性を残したままでモデル化）しておく。以上を踏まえて既存の方法論に対して複数視点による状態の扱いを付加したオブジェクト指向分析プロセスを提案した。

4.3 考察

現状のオブジェクト指向分析・設計については、まだ

まだ未熟であり、今後進化していく方法論である。

汎用的な手法としての洗練化も必要であるが、実務への適用という観点では、部門ごとに特化した方法論としての充実が当面要求される。たとえば、社内における実務への適用においても、たとえばある部門では、新たな図式を追加し、別の部門では作業順序を入れ替えたりしている。オブジェクト指向によるソフトウェア開発は現場にとっては、大きなカルチャ・ショックである。実務ではさまざまなやり方でソフトウェアを開発している。そこにおいては数多くのノウハウが培われ活用されている。そうしたノウハウは長年の体験から得られたものである。それらの蓄積を保ちつつどのようにオブジェクト指向方法論を導入していくかも大きな課題である。

5. CASE

5.1 CASEの課題

CASEというキーワードも一つの流行語であるが、ソフトウェア開発を計算機で支援しようという動きはソフトウェア・エンジニアリングが認識された当時からあった。ここに来て注目を集めている背景は、GUI環境を備えたWSの普及であろう。国内においても、米国から10年遅れて、構造化分析／設計が活用されはじめ、上流工程に対する認識が高まってきたことから、上流CASEに対する期待観が強くなった。しかしながら、上流CASEの利用の実態としては、お絵描きツールあるいは消書ツールの域を越えていないといっても良い。その原因としては、利用者側の方法論に対する理解度の低さもさることながら、最大の要因はツール側の支援機能の低さである。さらに、異なるCASEツール間のデータの互換性も重要な問題である。これについては、最近の話題として、開放型CASEとして、様々なCASEツール間でのデータ変換が行えるようにする

動きが活発であり、一つの課題がクリアできそうである。

ここで特に強調したいのは、CASEによる方法論の支援、言い換えれば手順の誘導（ナビゲーション）である。オブジェクト指向に基づくソフトウェア開発手順としては買い物メモアプローチ、ラウンドトリップといったいわゆるプロトタイピング手法である。これは現実的に最初から適切なクラスライブラリを構築することは難しく、どうしても試行錯誤的に構築せざるを得ないためである。プロトタイピングという言葉の持つ意味から、戦略もなくできたところから動かして確認することを繰り返すような開発手順を連想する。しかし、実務の大規模システムを対象とした場合には、このような開発手順ではプロジェクト管理の側面からも不都合が生じる。このような状況では、手順が標準化、可視化されているとは言い難い。

もう一つ強調しておきたいことは検証系の充実である。CASEの目的からすれば、そこで記述された内容の吟味を計算機で支援できることは当然のことである。

そこで、これらに対する取り組みを以下に示す。

5.2 CASEへの取り組み

5.2.1 手順の標準化

CASEでの手順の支援は、ナビゲーション機能、ガイダンス機能と呼ばれる。筆者は、このナビゲーション機能の実現を目指して情報処理振興事業協会の小高らとオブジェクト指向分析プロセスを定式化中である[Honiden93]。そこでは、3つの例題に対して、詳細な分析を実施すると同時に、後の解析のために分析ログを記録した。そして、詳細な仕様化プロセスを構築することを試みた。その仕様化プロセスは次の4つの要素から構成される。

(a)段階的に詳細化された分析プロセス

(b)分析項目における作業時に用いられるノウハウ

(c)作業間のデータの使われ方を示すデータフロー・モデル

(d)作業間の順序関係を示すコントロールフロー・モデル
例として図1は全体のデータフロー・モデルを示す。

データフローには、以下の3種類が存在する。

- ・連続プロセス間順方向データフロー

(図1：矢印1)

コントロールフロー上、次に実行されるプロセスにおいて、当該データを用いる場合。

- ・不連続プロセス間順方向データフロー

(図1：矢印2)

時間的に不連続ではあるが、データの生成時間とプロセスの実行時間の関係の整合性がとれている場合。

- ・逆方向データフロー

(図1：矢印3)

データの生成順とプロセスの（期待されている）実行順が逆転している場合。この場合、データの生成をトリガとする作業の後戻りが発生する。

一方、コントロールフローとして、次の4種類のコントロールフローを抽出した。

- ・前向きコントロールフロー

分析の基本的な作業順序。

- ・データ生成遅延による後向きコントロールフロー

属性やサービスの定義においてそれらを持たないクラスが発見されたときに発生する。

- ・作業の失敗による後向きコントロールフロー

分析者のミスなどの理由により発生する作業の後戻りを表す。

- ・既出データの検査のための後向きコントロールフロー

以前に生成したデータの正当性の検証および補強が可能となるようなコントロールフローを表す

以上示した考察をもとに現在ナビゲーション機能を実装中である。さらに、多くの適用評価を繰り返すことによって洗練化すると同時に、一般的なノウハウの蓄積も残っている。

5.2.2 代数的仕様を核言語とするオブジェクト指向CASE

オブジェクト指向分析・設計の作業プロダクトを代数的仕様として保持し、分析・設計作業の正しさの検証、これらの情報をまとめた仕様書の生成、さらにプロトタイプの生成などを支援する環境を構築中である[山本92]。オブジェクト指向分析・設計では、その表現手法としてデータフロー図、ER図、状態遷移図を有している。筆者らはそれらの図で記述された内容からオブジェクト指向代数的仕様言語(OS)に変換する(図2参照)。OSは代数的仕様にオブジェクト指向の概念を取り入れた仕様記述言語であり、オブジェクト指向としての特徴はクラス単位の仕様と継承概念の導入である。OSに変換することによって、設計者の意図に合致しているかどうかの仕様実行および検証項目による正しさの検証する定理証明による自動検証手段が用意されている。さらに、自動検証によって正しさの検証された分析・設計結果からC++プログラムを生成する。ここでのポイントはオブジェクト指向分析・設計においては様々な図を用いて情報が入力されるが、それらの情報を統一的に論理検証していることである。

5.3 考察

ここで紹介したのはナビゲーション機能と検証系に関する研究であるが、CASEにおいては、部品化再利用、リポジトリ、リバース・エンジニアリングなど、実用化を

促進する上では必要な機能が多岐にわたっている。従ってこれらに関する実用化研究も必要である。ある意味で実用化の観点では、CASEの諸機能の充実よりも、その背景にあるソフトウェア開発方法論の普及、教育および、異なるCASE間のデータの互換性（開放型CASE）の実現の方が重要であると言える。次世代のCASEにおいては、ナビゲーション機能は必要であろうが、研究期間がもう少し必要である。ただ、地道な努力が実用化に結びつくと考えている。一方、検証系については、記述性と検証可能性の間にトレード・オフがある。これを解決するには、検証能力が保証されたいくつかの有効な仕様記述手法の組み合わせが、実用上、現実的な解決策ではあるがやはり限界がある。基礎研究の促進が要求されるところである。

6. むすび

本稿では、種々の上流工程の支援手法の産業界における実用化に向けての取り組み事例として、筆者らのアプローチを紹介した。いずれのアプローチにおいてもすぐに役立つ技術ではないが、実用化に向けて鋭意努力を続けていかなければならないと思っている。

最後に産業界において、ソフトウェア・エンジニアリングの実用化に向けて研究を遂行している担当者として、日頃感じていることを述べて筆を置きたいと思う。筆者は、基礎研究の成果（ソフトウェア科学）をどのようにソフトウェア・エンジニアリングに適用できるかという観点での研究を行っている。そこで痛感していることは、基礎研究と実用化研究の隔たりであり、しかも、その隔たりを埋めるべく研究（応用研究であろうか）が国内外を見渡しても極めて少ないことである。基礎研究者は実務ソフトウェアに対して、また実用化を目指している担

当者は、逆に基礎研究の成果に対してもっと目を向けるべきである。すなわち、産業界の立場としては、基礎研究の進歩を常に睨みながら、使える技術をどん欲に取り組んでいく姿勢が必要である。また、産業界においても、すぐには役に立たなくても、将来、役に立つことが期待される基礎研究に対してもう少し積極的に取り組むべきであると考える。

謝 辞

日頃ご指導いただく（株）東芝 研究開発センター システム・ソフトウェア生産技術研究所の西島誠一所長、新井政彦部長、大筆豊研究主幹に感謝いたします。
なお、本稿の一部は第5世代コンピュータプロジェクトの一環として行われたものである。

参考文献

- [本位田92a]本位田真一、大須賀昭彦、内平直志、代数的仕様と時制論理によるリアルタイムSAとオブジェクト指向設計の融合手法、情報処理学会論文誌
vol. 33, no. 2, 1992
- [本位田92b]本位田真一、小高信人、岸本芳典、知的ソフトウェア開発環境を目指す、コンピュータ科学、Vol.2、No. 4、1992
- [Honiden93]Honiden, S., Kotaka, N., Kishimoto, Y.,
Formalizing Specification Modeling in OOA,
IEEE Software, Vol. 10, No. 1, 1993
- [山本92]山本純一、大須賀昭彦、本位田真一、代数仕様を核言語としたオブジェクト指向分析・設計支援、情報処理学会、ソフトウェア工学会研究会資料89-4、1992
- [入江92]入江豊、斉藤悦生、仕様・運用変更を意識した複数視点によるOOA、電子情報通信学会、信学技報KBSE92-25、1992

[山城92]山城明宏、中野裕子、本位田真一、OOAとリアルタイムSAの変更容易性に対する考察、情報処理学会、ソフトウェア工学会資料89-6、1992

[浦岡92]浦岡徹、山本純一、大須賀昭彦、本位田真一、代数仕様によるプラント制御エキスパートシステムの記述と検証、情報処理学会、ソフトウェア工学会資料86-15、1992

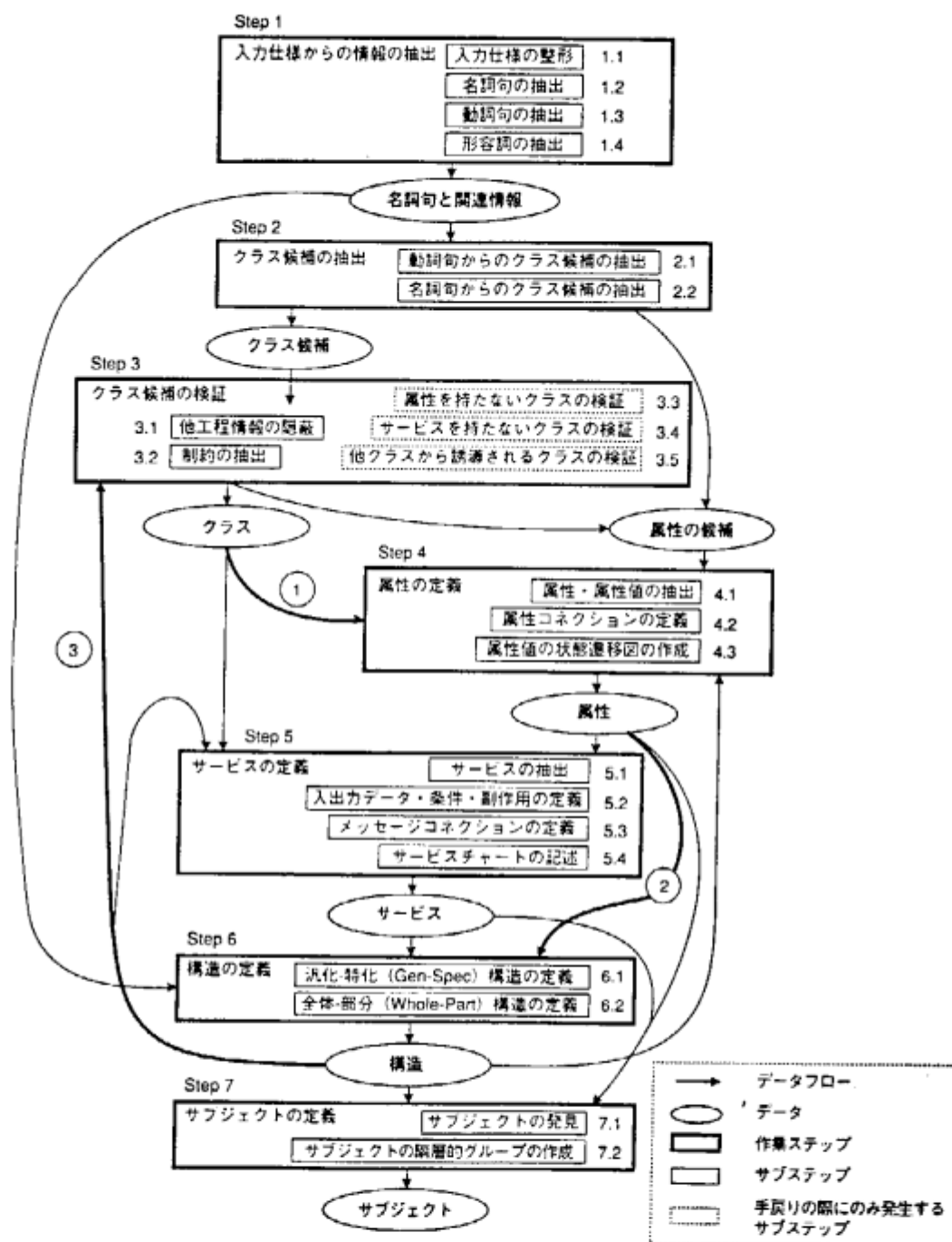


図1 全体データフロー OOAプロセス・モデルにおいて、作業間のデータの使われ方を示したものである。

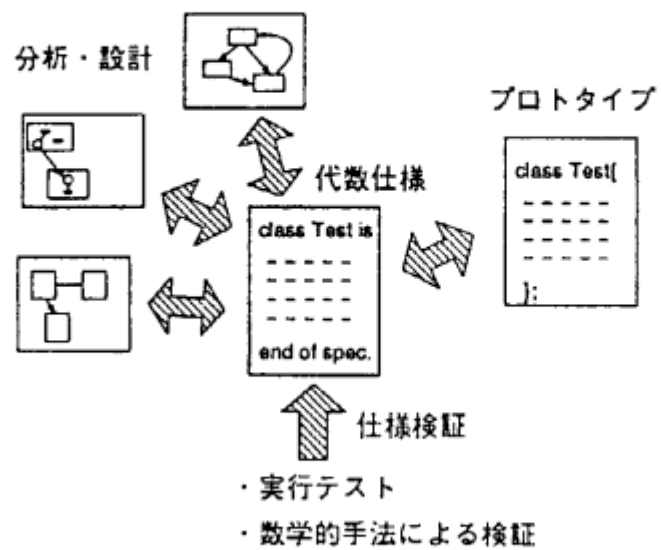


図2 オブジェクト指向ソフトウェア開発環境の構想