

システム開発方法論の構造的展開

芳賀 正憲

1. はじめに

システム開発の方法論は、もともと非常に広範な概念を含んでいるが、産業界では一般にこれを業務手順と技法に二分して整理している。

システム開発業務は、課題に対する1つのsolutionとみなされるが、上記のような方法論の整理は、問題解決技法の体系化の進め方と対比して考察すると理解しやすい。

例えば、情報処理振興事業協会等が編集した「問題解決技法」のテキストによると、問題は、図1に示すように、プロセスとこれに対応する技法によって解決が図られる。このとき、プロセスと技法の関係には、図2の左に示すように、プロセス毎にこれを支える技法が並立して存在する場合と、右のように、まず問題解決のための大きな技法があって、その中にその技法を実行する手順としてのプロセスが存在する場合の2つのケースがある。

システム開発の場合、右のように、まず開発技法があって、次にその技法を実現する手順が存在するという枠組みが妥当ではないかとも考えられる。しかし、技法の多様性や、組織により選択される技法が異なること、1つの技法についても年々発展していくことから、システム開発にとって必要な業務の手順と対応する技法を、便宜的に左のように分離・独立させて整理する考え方が一般的になったと思われる。

本稿では開発方法論のうち、まず技法について、基本的な考え方の変遷を考察する。次に、最近提案された共通フレームを中心に、業務手順の標準化がどのような形で行われているかを述べ、その構造が、技法の最新の考え方に対応したものであることを示す。最後

にこれからの大きな課題として、開発生産性や品質を向上させるため採用されている方法論の特徴について言及する。

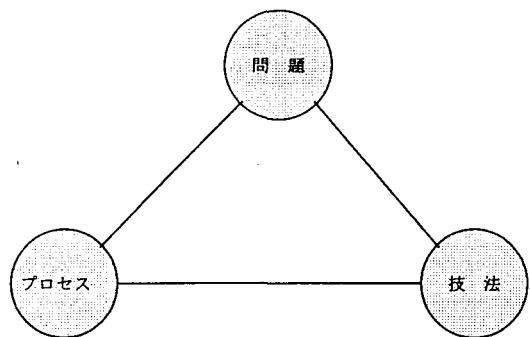


図1 問題解決におけるプロセスと技法

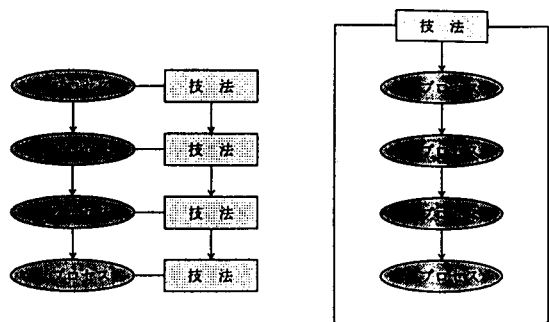


図2 プロセスと技法の関係

2. システム開発技法の発展

2.1 DFD技法の登場

システム開発の方法論というときのシステムとは、最終的にコンピュータ上のプログラムを含むとしても第一義的には、人間の組織的な業務活動を意味している。

はが まさのり 新日鉄情報通信システム(株)
〒104 中央区新川2-20-15

システム開発の技法とは、このような人間の組織的な業務活動の現状を分析し、改善された新しい活動の形態をデザインする方法である。このとき、分析とデザインをいかに構造的に進めていくかということが、技法発展上の大きな課題となっている。

このような技法は構造化技法と呼ばれているが、その歴史は“5W1H”に分けて考察すると分かりやすい。すなわち、why（なぜ）、who（誰が、どのコンピュータが）、when（どのようなタイミング、順序で）、where（どこに設置されて、どこを対象に対して）、what（どのような世界を対象にして）、how（どのような処理をするか）の6項目である。

構造化技法の歴史は、まずhowを明確化することから始まった。最も普及が進んでいるのは、Data Flow Diagram (DFD) 技法であり、1978年DeMarcoによって提案されたものが有名である。図3に、DFDの1例を示す。

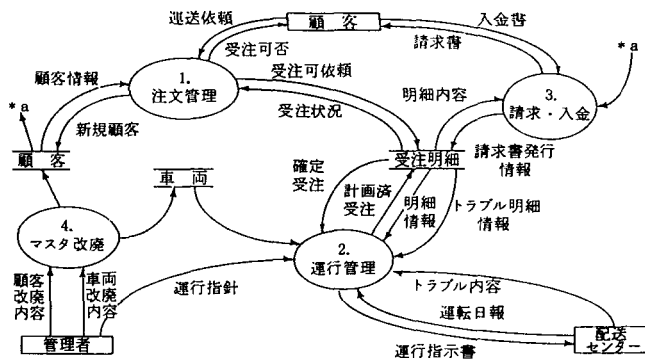


図3 Data Flow Diagram の例 (K 運輸会社現行論理)

情報処理の世界は、基本的に情報の伝達、変換、蓄積から成り立っている。このとき、伝達を矢印、変換を円、蓄積を（電気のコデンサにならって）横線2本で表現したものがDFDである。つまり、DFDは情報処理の世界を基本要素に分けて的確に図示することが可能であるがゆえに、必然性をもった分析技法であるといえる。

DeMarcoのDFD技法の意義は、単に情報処理の構造的な表現法を示しただけでなく、新しいシステムを作っていくためのプロセスを提案したところにも存在する。彼によれば、新しいシステム仕様を決定するまでのプロセスは、現行物理→現行論理→将来論理→将来物理（の各モデルの作成）というステップをたどる。

2.2 最新構造化技法の確立

DFDを中心とする構造化技法は、米国を中心としてソフトウェア・クライシスに悩むシステム開発部門に大きな期待をもって迎えられたが、短期間にいくつかの問題点が指摘されるようになった。これらについては、解決策が次々に提案され、最新構造化技法として体系化が進んでいる。その主要なものは、次のとおりである。

(1) 論理化は、DFD技法の中でも中核となるプロセスであるが、その手順の標準化がむずかしい。

これに対してはMcMenaminとPalmerが、イベントとこれに対応する処理を定義するエッセンシャル（本質）モデルの考え方を提案し、解を与えた。

(2) DFD技法のプロセスとして示された4段階の手順を忠実にたどると、非常に時間がかかる。

これに対しては、最初から将来論理に相当するエッセンシャル・モデルを作成し、それをもとに将来物理を決定するという、簡略化された手順が提案されている。ただし、これが実行できるのは、システムエンジニアが当該業務を熟知している場合に限るとされている。

(3) DFD技法では、制御やタイミングの表現が困難である。

DeMarco自身は、タイムサイクルやタイミングの表現は、システムの機能を示すに当たって、本質的なことではないといっている。しかし、リアルタイムのシステムにとっては、制御やタイミングは系の同定のため欠くことのできない要件である。

この課題の解決はHatley、Wardらによってなされたが、例えばHatleyは、DFD中のプロセスに対する制御信号の流れを示すControl Flow Diagram (CFD)とコントロールの仕様を表す状態遷移図を導入している。

図4に、図書館システムにおける図書の状態遷移の表現例を示す。

(4) これまでの技法は、処理を中心にシステムを分析し、データを各機能の付属物のように扱ってきたが、システムの規模が増大するに伴い、同一内容のデータが各機能毎に別々に定義をされたりメンテナンスされたりして、整合性の確保が困難になってきた。

これに対しては、次のような考え方で解決が図られ

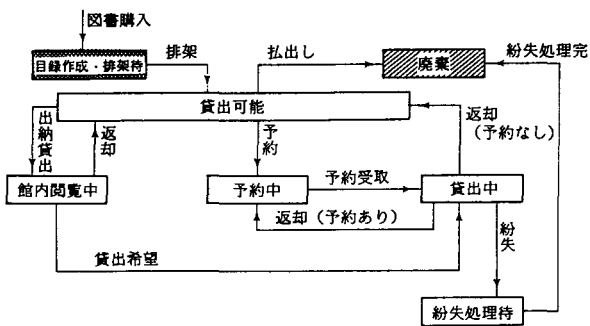


図4 図書館システムにおける図書の状態遷移の表現例

ている。

一般的に処理機能は、ニーズの変化に伴い常に变化するが、データで代表されるシステム化の対象自体は処理機能に比べてより安定的である。

そこで、データ構造自体を重複や不整合のない形に定義しておき（これをもとにファイルやデータベースを設計する）、そのあとで状態変化や処理機能を決定していくほうが、より構造的に優れたシステムができあがるのではないかと考えられるようになった。

データ構造を表現する技法として広く採用されているのが、Entity-Relationship Diagram (ERD) 技法である。図5に、卸問屋業務のERDによる表現例を示す。

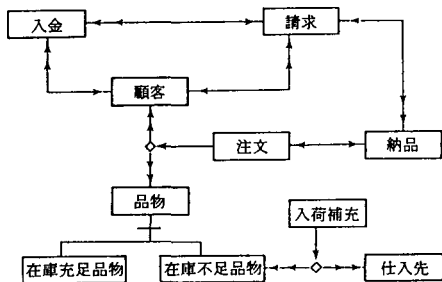


図5 卸問屋業務のERDによる表現例

2.3 構造化技法の今後の展開

DFDは、「5W1H」のうち、howを表現するものであり、またCFDや状態遷移図はwhenを、ERDはwhatを表現していると思なすことができる。このとき、whoとwhereは自明（例えば中央計算機室のメインフレームが工場のオペレータの実行業務をサポート）のこととされていた。しかし、最近のように、ワークステーショ

ンやパソコンの能力が向上しネットワークが拡大して分散システムが現実化してくると、whoとwhereを含めて分析を行う5次元の構造化技法の確立が緊急の課題となってきている。また、5次元の各要素に対して、ニーズと技術的制約の両面からwhyをはっきりと示すことのできる基本原理を明らかにすることも、重要な研究テーマとして残っている。

3. 業務手順の標準化—共通フレームの制定

3.1 共通フレームの制定

1994年1月、情報処理振興事業協会より「ソフトウェアを中心としたシステムの取引に関する共通フレーム」が発表された。ここで共通フレームとは、システム開発市場の健全化を図るため、開発に関連した作業群を俯瞰し、可視化した共通の枠組みであり、コンピュータメーカ、情報サービス企業、ユーザ企業、学界の代表者で構成される検討委員会が策定されたものである。

共通フレームでは業務手順をいわゆるWork Breakdown Structureで整理している。大分類として購入、供給、企画、開発、運用、保守、管理、環境整備、教育訓練、文書作成、構成管理、品質保証、問題解決、組織の確立・評価・改善、システム監査の15プロセスが定義されている。中分類はアクティビティと名づけられ、例えば開発プロセスは13、運用プロセスは7つのアクティビティから成り立っている。各アクティビティは、いくつかのタスクに分けられ、これがWork Breakdown Structureの最小単位となっている。

各タスクについて、入力・出力となるドキュメント様式、入力から出力への変換手順、変換に際し用いる技法・ノウハウ、出力の評価関数・チェックポイントなどを整理すれば、システム開発における業務手順は細部に至るまで明確に表現できることになる。

共通フレームで提示されているプロセス、アクティビティ、タスクは、必要に応じて取捨選択したり、繰り返し実行したり、複数のものを持って実行してもよいことが明記されている。このような作業は、最近テラリングと呼ばれている。

現実の開発するシステムは、プロジェクトによって業種・業務、規模、プラットフォーム、再利用可能な資産、開発メンバーの既存知識のレベル等を異にしている。それらに応じて適切にテラリングを行うこと

は、プロジェクトを成功に導くためにきわめて重要な活動である。

共通フレームで注目すべきことは、プロセス、アクティビティ、タスクを、X軸/Y軸の2次元座標に展開すべきことが提案されていることである。ここでX軸は、実行のタイミング、順序を示す時間軸であり、Y軸は、その業務が何を対象に行われるかという、対象を表す軸である。一般に業務を行うに当たっては、その対象に応じて必要な技術や資源が決定され、組織や役割分担が決まってくる。したがって対象軸は、その業務の実行組織を示していると見ることができる。

共通フレームでは、時間軸/対象軸による整理の重要性は指摘されているが、細目の決定は各開発組織に任されている。現実の開発を実行するには、当然これらを明確にすることが必要である。

3.2 開発業務標準の事例

上に述べた共通フレームに先行し、かつ共通フレームと同等の概念で成り立っているのが、富士通(株)の開発業務標準SDEM90である。SDEM90ではWork Breakdown Structureだけでなく、時間軸/対象軸とも定義されている。以下に、その構造を紹介させて頂く。

SDEM90では、業務の対象をカテゴリと名づけ、時間軸となる工程より優先して提示している。

カテゴリとしては、まず大きく実世界とコンピュータ世界、両者間のインタフェース、開発資源補給世界の4世界を考え、次にこれらを業務、ソフトウェア、ハードウェア、システム仕様、開発支援、プロジェクト管理の6対象に分類している。さらにこれらを、23の小カテゴリに細分化している。例えばシステム仕様は、システム機能、データ構造、性能、信頼性・セキュリティ、運用・保守、移行の6つの小カテゴリから成り立っている。

時間軸は、システム企画から保守・システム評価に至る11の工程に分かれている。

対象軸を23、時間軸を11に分けることにより、253の要素をもつマトリクスができあがる。開発に関係する業務は、すべてこのマトリクス上に展開することができる(ただし、共通フレームにある購入、供給など一部のプロセスは除く)。

このマトリクスにより、通常V字型で表現される、システム企画から運用テストまでの主要な開発業務以外に、多くの重要な作業が、各カテゴリ、各工程毎に

行われる必要があることが分かる。

対象軸が役割分担に対応していることを考えると、このマトリクスの行を見ることにより、ある部署が工程毎に実行すべき業務が分かり、列を見ることによりある工程では各部署が何をなさねばならないか、一覧的に把握ができる。

マトリクス上に位置づけられた各業務については、Work Breakdown Structureの他、作業のポイントや技法/ツール、ノウハウ事例が提示されている。

3.3 共通フレームと最新構造化分析

共通フレーム及びそれに先行して提案されたSDEM90は、対象軸/時間軸の2次元座標上に展開された業務手順から成り立っている。

前述のように最新構造化分析では、システム開発の対象とする業務活動を、what、when、howの3次元で分析する。ここでwhatを対象軸、whenを時間軸、howを業務手順に対応させると、共通フレーム及びSDEM90の構造は、最新構造化分析を、ユーザの業務ではなく自らの開発活動そのものに適用して得られた結果であることが分かる。また、SDEM90でカテゴリが重要視されているのは、最近のデータ中心設計やオブジェクト指向の動きに対応したものであることが理解される。

4. 生産性向上への取り組み

価格破壊が叫ばれる中で、システム開発の生産性についても大幅な向上が要請されている。

生産性を従来より大きく向上させた開発事例が、最近いくつかの文献で紹介されている。それらのプロジェクトの特徴として、次のような項目が挙げられる。

- (1) エンドユーザとSEが一体となった少人数のチーム編成
- (2) 上記の編成で、全工程一貫開発
- (3) 大きなシステムは、上記の編成チームが複数で並行開発
- (4) 開発ドキュメントの簡略化
- (5) プロトタイピングとスパイラルアップ方式の採用
- (6) データ中心アプローチ
- (7) RDB、4GL、CASEツールの採用

技法としては、従来から生産性や品質の向上に役立

つとされていた方法を、複数組み合わせる相乗効果を挙げている。その上に、プロジェクト編成上の工夫でドキュメント作成とコミュニケーションのための負荷を軽減している。

これらの特徴を見ると、生産性向上のために実行されていることに必ずしもむずかしい概念はなく、すべて既知のものである。その意味で、われわれの従来の開発方法論改善のアプローチは基本的に今後も生かすことができる。

しかし、生産性向上という目的達成のため、これらの諸概念が高度に統合された上、今まで管理水準の確保のため実行されていたドキュメント整備などの諸作業が簡略化されている。

この方法は、ある意味においては20数年前の、システム開発の初期段階に帰った状態といえる。当時に比べ複雑度の増した開発環境のもとで、20数年前と類似した開発スタイルを採用することには、大変な管理上のリスクが伴う。このリスク回避のためには、生産性と品質の向上を同時に担保する開発プロセスのメカニズムの一層の解明と、これを確実に実行しうるシステムエンジニアの育成が必要である。

5. おわりに

システム開発の方法論に関し、技法の発展、業務手順の標準化、生産性向上施策の動向について述べた。

100万人を超えるといわれるシステム開発技術者が今後これらの諸技術について、共通の認識をもって実践を積み重ねながら、業務基盤の確立に努めていきたいものである。それとともに、これらの諸技術を総合的にサポートするツールの発展が切実に望まれる。

本稿に対し、大方のご批判が頂ければ幸甚である。

参考文献

- [1] 情報処理振興事業協会他：問題解決技法テキスト，東京計算サービス（株）（1991）
- [2] T. DeMarco（著），高梨他（監訳）：構造化分析とシステム仕様，日経BP社（1990）
- [3] S. M. McMenamin 他：Essential Systems Analysis，Yourdon Press（1984）
- [4] E. Yourdon（著），荒川（訳）：CASE時代の最新プロジェクト管理技術，マグロウヒル（1990）
- [5] D. J. Hatley他（著），立田（監訳）：リアルタイム・システムの構造化分析，日経BP社（1989）
- [6] P. Chen: The Entity-Relationship Model - Toward a Unified View of Data ; ACM Transactions on Database Systems , Vol. 1 , No. 1 , pp. 9-36（1976）
- [7] E. Yourdon : 見直しの時期にきた構造化分析技法 ; 日経コンピュータ , No. 126 , pp. 105-110（1986）
- [8] 芳賀：ビジネス情報システムの開発技術；システム／制御／情報，Vol. 37, No. 3, pp. 24-31（1993）
- [9] 大野（監修），共通フレーム検討委員会（編）システム開発取引の共通フレーム，通産資料調査会（1994）
- [10] 村上：ソフトウェアのライフサイクル管理；情報処理，Vol. 33, No. 8, pp. 912-921（1992）
- [11] 佐藤：クライアント／サーバデータベース設計テクニック，ソフト・リサーチ・センター（1993）
- [12] 渡辺：ソフト生産性は3倍になる；日経コンピュータ，No. 340, pp. 66-79（1994）