

亀の子モデルを適用した会計処理システムの要求分析法

システム開発における要求分析の重要性は古くから言われ、ウォーターフォールモデル、プロトタイピング、スパイラル方式、Extreme Programming、UMLなどがある。しかし、これらの手法はシステムエンジニア同士間での成果物の表現方法で、エンドユーザには馴染み難く結局はプログラムが出来上がる迄、何が出来るか分からないのが実態である。今回、会計システム開発の要求分析段階で要望を出すエンドユーザと、その要望を受けるシステム開発者の双方とも理解ができる技法、「亀の子モデル」を考案し開発を行った結果、この技法が有効であることが実証された。この技法は1対の勘定科目をEntity、関係をRelationshipとするEntity Relationship Diagram表現に基づき、データ関係と処理内容を一体記述する。

Requirement Analysis Method for Accounting Procedure System Applying Tortoise-shell (Hexagonal) Pattern Model

The importance of requirement analysis in the area of system development has long been discussed, and consequently people have devised Waterfall Model, Prototyping, Spiral Method, Extreme Programming and UML. However, these techniques are presentation methods for their products among fellow system engineers, which are so unfamiliar for end users to understand that eventually people can not understand the contents until the program is completed.

We have recently devised and developed a technique named "Tortoise-shell (hexagonal) Pattern Model" that is understandable for both the end users who make requests and the system developers who receive the requests at the stage of requirement analysis in system development. As a result, it was verified to be effective. This is a method representing the data and its processing procedure in an integrated form, Entity-Relationship Diagram, comprising a pair of accounting subjects as Entities and the relationship between them Relationship.

1. はじめに

システム開発フェーズにおける要求分析の重要性については昔も今も変わりなく、寧ろ、近年、情報システムの適用範囲が非常に広範囲で、適用される専門分野も深い処までになるにつれ、その重要性は重みを急速に増している。

渡瀬 浩一 Kouichi Watase[†] 山谷 茂 Shigeru Yamaya[‡]

[†]旭電化工業株式会社 Asahi Denka CO.,Ltd

[‡](有)システムコーディネート System Coordinate, Inc.

E-mail: [†] watase@adk.co.jp, [‡] syamaya@soleil.ocn.ne.jp

情報処理の内容が複雑になる一方、従来の処理形態（オンライン・インプット⇒バッチ処理）から、全てをオンラインリアルタイム処理する要求が急速に高まってきたこと、また、システム開発そのもののスピードアップが要求され、開発期間が年々短縮を余儀なくされてきている。このようなことから、要求分析手法自体も、開発対象となるシステムに対応して提唱されている。

今回、会計業務システム開発に当り要求仕様を出すユーザとその要求仕様を分析するシステム開発者がお互いに理解ができる技法「亀の子技法」を考案し、これを活用して開発を行った。

2. 開発における要求分析の重要性

オーソドックスなシステム開発プロセスとして、図-1にウォーターフォールモデルを示す。ウォーターフォールモデルは、その後、ライフサイクル有害説⁽¹⁾等の批判があり、代わってプロトタイピング⁽²⁾やスパイラル⁽³⁾などが提唱されてきたが、ウォーターフォールモデルの中にある開発フェーズの区分は依然として変わってはいない。

ウォーターフォールモデルの最大の欠点は、プログラム製造・試験が終了しないと、果たしてユーザの要求通りに製作されたかどうかの判断ができないことである。

ユーザ要件のとり違いがあると、開発の最終工程で膨大な手戻りや手直し作業が発生する。かつて大規模システムをウォーターフォールモデルで開発し、多大な手直し作業が発生したケースが起きたことで、ウォーターフォールモデルは悪者扱いされている。

しかし、ウォーターフォールモデルの分析フェーズに対する重要性が変わった訳ではない。

プロトタイプやスパイラルモデル、最近の動向として注目されているソフトウェア開発プロセス、Extreme Programming⁽⁴⁾が目指すのは、やはりユーザ要件を如何に素早く確認し、手戻りを少なくして、開発プロセス全体の工数を低減することである。

このことはBarry W. Boehmがウォーターフォールモデルで示した要求分析フェーズがシステム開発プロセスでの最大のポイントであることを示している。

3. 従来の一般的要求分析法

3-1. 業務分析手法

上述のように、要求分析フェーズがシステム開発において最も重要であることが認識されて数々の手法が提案された。古くは機能情報関連図を用いた機能と情報の関係で、業務の流れに沿って分析を行った。その後構造的に分析しようとしてSADTが登場し、DFD(Data Flow Diagram)に発展している。DFDの例を、図-2に示す。DFDはデータベース設計へ連動し、CASEツールによってサポートされており、最近でも、EA(Enterprise Architecture)策定ガイドライン⁽⁵⁾の中でも業務分析を行う手法として紹介されている。

3-2. UML

オブジェクト指向が盛んになって、UMLが急速に広がってきている。これは、James Rumbaugh, Grady Boochの第2世代のOMT(Object Modeling Technique)に、Ivar Jacobsonが加わってUML(Unified Modeling Language)⁽⁶⁾として作り上げた物で、

クラス図、シーケンス図、コラボレーション図、状態図、アクティビティ図などからなるもので、UMLは概念モデル、分析モデル、設計モデルの各ステップをサポートしており、CASEツールも出ており、業務分析としてはシーケンス図が用いられている。

3-3. 制御系手法

制御系システムでは、業務系システムで用いられているDFDでは分析することは難しく、状態遷移図を中心とした技法が用いられている。状態遷移図は制御を行う流れを分析する手段の為に、最適な技法として提案された技法である。これにはR. ネットやペトリネットなどがある。

3-4. 従来の分析手法の問題点

しかし、上述した従来の手法はシステムエンジニア同士の分析成果物の表現方法であって、エンドユーザが、その成果物を見て、理解して積極的に意見を述べたり、要望事項の再確認をしたり、自分自身の業務改革点を把握することは、殆どの場合不可能である。

また、データ中心分析では、THデータモデル⁽⁶⁾、T字形ER手法⁽⁷⁾など示唆に富む技法が発表されており要求分析に当り参考になるが、上述と同様にエンドユーザには理解し難い成果物である。更に、もう1つは、作業しているシステムエンジニアを見てみると、現状に肯定的な成果物の作成に陥り易い面があることである。それは現行システムのデータに拘泥し過ぎ先入観を持つことになる。この結果、業務の本質を見失い、改善はできても、飛躍した業務改革を行うことは、出来なくなりがちである。

従来の手法は、提唱者にとって意外かもしれないが、何れもシステムエンジニアにとっても習得が難しく、エンドユーザが自分で描くなど思いもよらない技法である。

4. 会計処理－貸借モデル－

会計に関わるシステムを開発する場合にユーザ側が要件を説明する場合は、常に貸借モデルで行われる。そして、このモデルを使用した「仕訳」が焦点になる。

4-1. 経理担当者が使う仕訳（貸借モデル）の普遍性

現在の複式簿記の起源は、15世紀のベニスの商人の帳簿から始まったといわれており、1494年にイタリアのパチョーリによって簿記の本が書かれたのが由来とされている。それ以来、基本ルール（借方勘定の合計と貸方勘定の合計は必ず一致する）は変わっていない。このモデルは、現在の経理業務においても、表現することができる普遍の原理である。その表の表現形式はT字形で、左に借方勘定、右側に貸方勘定を記述し1つの取引を貸借二面から捉えるやり方である。簡単な例を図-3に示す。この伝統的なやり方で、経理担当者は簿記をつけ、残高試算表を作成する際も貸借勘定に集約するなどしており、経理担当者が経理業務を説明するのに必須のものである。

4-2. システムエンジニアにとっての経理処理の難しさ

システムエンジニアが財務会計処理の原理図を書くと、図-4のようになるのが一般的である。仕訳ロジックそのものは、プログラムの詳細設計時点で、プログラム仕様として捉えようとする。従って、この要求分析定義フェーズでは経理担当者とシステムエン

エンジニアの間に乖離は無い。しかし、これは、両者の考えが一致したのではなく、最重要課題を先送りしたに過ぎない。つまり要求分析・定義フェーズでは経理処理システムの仕様仕様の打合せはまだ始まっていない。多くの場合、この後、システムエンジニアの泥沼的仕様固め作業が続くのである。

5. 亀の子モデル

初めて貸借勘定に直面するシステムエンジニアにとって、貸借モデルでの訓練は、経理アレルギーに成りかねない。一方、経理担当者側は複式簿記仕訳以外の方法では説明が不可能である。経理担当者は、システムエンジニアがどんなデータベースを考え、どんな処理を考えているのか、を問題としているのではなく、自分の考えるタイミングで、処理されてほしい帳簿が出力されるであろうかを心配し、またシステムエンジニアは、あの分かり難い仕訳処理の設計内容が確認できるのか不安を持っている。要求分析手法としてこの双方の悩みを一举に解決するのが、今回の亀の子モデルである。

5-1. 亀の子モデルとは

本モデルは、前述の貸借モデルを分析して2つの事項に注目して考案したものである。第1は、図-3に見られるように、一般の簿記では、借方と貸方の行数が合っていない場合がしばしばあるが、空白の行に対して勘定科目を補うことで、1対の勘定科目に仕訳けられる。

第2は、総勘定元帳の作成に必要なデータは勘定科目毎の前残、当残と、増減のデータで、補助元帳作成に必要なデータは、上記の外には個々の明細情報だけである。

つまり科目別の金額増減台帳（前残金額、増積算金額、減積算金額、現在残高金額を持つデータファイル）と、明細伝票（個々のトランザクションのデータファイル）の2種類のデータファイルで、全ての会計処理を表現可能とするのが亀の子モデルの出発点である。「亀の子」は、データモデルのリレーションシップの表現を原型としている。「亀の子」で、リレーションシップ・タイプのエンティティを表現するに至った過程は、上述したように、貸借1対の勘定科目が最小明細単位である事を見つけた分析作業からきたものである。

ここで改めて整理すると、亀の子モデルには次の3つの特徴がある。

- ①データの関係を示すことで、概念モデルの域を出て論理データ構造を明らかにできる。
- ②処理内容を記述できる。
- ③経理担当者が自分で処理内容を同一図上に描くことができる。

即ち、システムエンジニアと経理担当者がデータベースの構造と処理内容の記述作業を共同で作業できることが最大の特徴である。

5-2. 亀の子モデルの描き方のルール

亀の子モデルで図-3を描いたものが図-5である。この図は2つの四角の箱とそれを繋ぐ六角形の箱で構成されている。四角の箱は上記の科目別の金額増減台帳で、最上段に勘定科目名称名を書き、前月残、当月増、当月減、現在残の項目を置く。この2つの上下の箱の間にある六角形の箱が上記の明細伝票で、借方と貸方の科目を書く「/」で区切る行を置きその下に金額を書くようになっている。つまり、例題方式で、具体的に

書くことで曖昧さを排除できる。この六角の形に因んで、亀の子モデルと命名した。

5-3. 記述方法

特に難しいルールはないが、次のように決めてある。

- ①二重線の六角形がイベントを表す。金額のマイナスは認めずマイナスにならないように貸借科目を配置する。
- ②実線の六角形は、イベントに伴って起きる副作用で、図-5では消費税の処理を書いている。
- ③借方の科目金額増減台帳を上側に、貸方の科目金額増減台帳を下側に書く。
- ④二重の四角形は外貨建てであることを示す。
- ⑤実線の四角形は円価である。
- ⑥1つの取引によって派生的に生じる仕訳処理の内、消費税については本体金額と切り離せない処理であるので、亀の子の両側を点線で囲ってある。これは実際の明細伝票（ファイル）のレコードでは、本体レコードのアイテムの1つとする事で、消費税の性格を明確に表現するようにしている。

5-4. 仕訳パターン

仕訳パターンと分析例及び記述例について述べる。

- ①1つの取引（イベント）が1つの仕訳になる。
単純な貸借の仕訳を亀の子モデルで記述し、夫々の残高試算表の更新ロジックを記述する。図-5では、経費（一般管理費）で何かを発注して請求書がきたので、支払を行う仕訳の例である。
- ②1つの取引（イベント）で、複数の仕訳になる。
図-5では、消費税の処理を加えている。
- ③1つの取引（イベント）は本来1つであるが、外貨建ての輸出販売などは、債権管理は外貨であるが、売上管理は円価である。入金の場合は、更に為替処理を必要とする。為替差額は、営業外収益か営業外費用への処理が続く。

6. まとめ

経理システム開発において「亀の子」モデルを使用する事で、経理担当者とシステムエンジニア、双方が同じ土俵で要求分析できることになり、双方とも誤解や理解不足に陥ることなく効率的に作業が進められ、大きな効果をもたらした。

今回は経理システム開発の要求分析ステップへの適用として述べたが、本技法は、要求分析段階だけでなくこの先の段階であるデータベースの設計、そして、業務処理の各プロセスのプログラム仕様設計段階まで、常にユーザと共に、同一図面で共同作業をできことから、更に大きな成果を上げてきた。今後、これについても発表を行いたい。

この技法は、会計処理システムにだけ限定するものではない。種々の情報システムの開発に十分適用可能であることから、これについても別の機会に明らかにしていきたい。尚、次のプログラム設計製作ステップでは、プログラム設計技法WCAP技法⁹⁾を適用したことで大幅な生産性の向上を実現し、開発全体工程の短縮化を実現した。

参考文献

- (1) ソフトウェアプロトタイピング 有沢誠 近代科学社 1986.7 p17
- (2) プロトタイピング RolandVonk 黒田純一郎訳 共立出版 1992.9
- (3) CASE概説 竹下享 共立出版 1990.6 p24
- (4) 情報処理 平鍋健児 2002 Vol.43 No.3 No.4
- (5) EA策定ガイドライン 日経コンピュータ/I Tプロフェッショナル特別編集版 2003.12
- (6) わかりやすいUML入門 樫山友一 日野泰臣著 2002.9
- (7) データ中心システムの概念モデル 椿正明 オーム社
- (8) T字形ERデータベース設計技法 佐藤正美 ソフトリサーチセンター
- (9) NAT-WCAP 技法 渡瀬浩一 山谷茂 電子情報通信学会 信学技報 DE2003

図-1 : ウォータフォールモデル

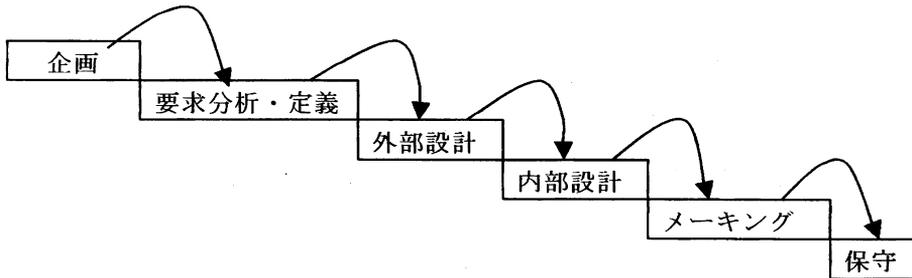


図-2 : DFDの簡単な例

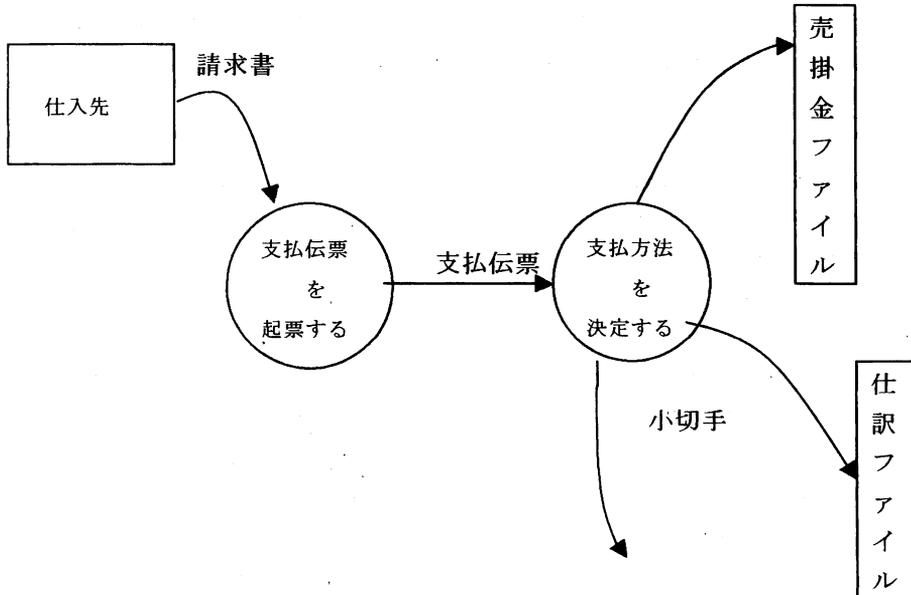


図-3：貸借の仕訳例

借方		貸方	
一般管理費	10,000円	当座預金	10,500円
仮払消費税	500円		
合計	10,500円	合計	10,500円

図-4：財務会計モデル

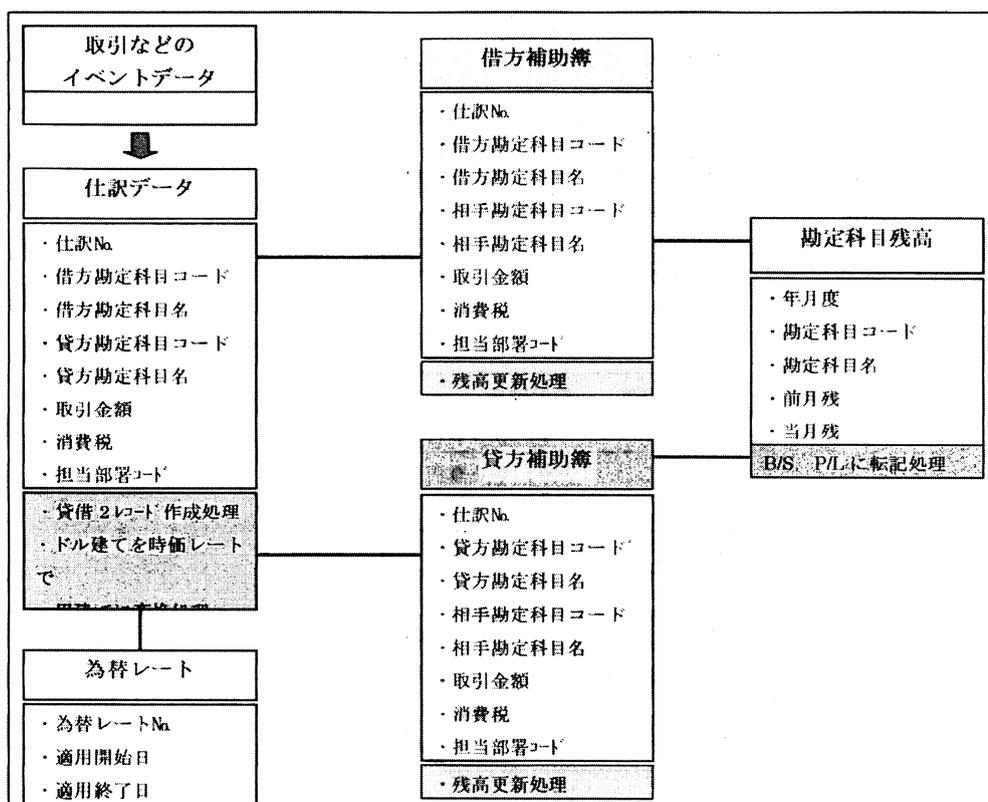


図-5：亀の子モデル

