

オブジェクト指向分析・設計手法の研究

3 J-1

清枝 嘉一（田中貴金属工業株式会社 システム部）

現在、企業の基幹システムの大部分は、開発パラダイムは構造化技術、開発プロセスはウォーターフォールモデル、言語はCOBOLで開発されている。その一方で各団体による研究・業務システムの開発等によりオブジェクト指向の有用性が証明されている。しかし、オブジェクト指向によるシステム開発を行うにあたり、「ユースケースからクラスを切り出すのが困難」・「クラス図とシーケンス図の整合性の確保が困難」などといった、分析・設計工程での効果的な設計手法の適用が困難であるとの問題を抱えている。

従来型の基本的なオブジェクト指向分析・設計手法は、①システム要件と仕様の確定、②ユースケースの作成、③詳細ユースケースの作成、④クラス図の作成、⑤シーケンス図の作成といった手順にて行われている。この従来型の手法で、業務モデルに対する分析・設計を行うにあたっての2つの問題について考察・問題の解決法を提示する。

問題の一点目は業務分析・設計工程におけるクラスの確定について、図が簡単に切り出すことができない場合が多々あるということで、「ユースケースからクラスを切り出す」、「クラス図とシーケンス図の整合性を確保する」といった事が困難であるといった内容である。

問題の二点目は、従来型の分析・設計手法はウォーターフォールのように、作業工程が順を追っての直列構成となっており、ある作業工程で仕様等の変更が生じた場合、前工程で確定したはずの仕様について変更を行わなければならない、その変更作業については大きな工数がかかることである。

一点目の、「クラスの確定が困難」であるという問題に対しての解決策として、従来の分析・設計方法のように、ユースケースから直接静的モデルであるクラス図の作成は行わず、インスタンスの動きを詳細かつ具体的に記述していく、動的モデルの一種である、シナリオ・インスタンス図の作成を行った。

（ここでは動的モデルとは、「シナリオ」・「インスタンス図」・「シーケンス図」・「状態遷移図」と定義する。）

シナリオはユースケースから具体的なオブジェクトの振る舞いに注目して、要求仕様から洗い出されたユースケースに対してシナリオを記述していく。更にインスタンスの動きを細かく具体的に記述し、1つのユースケースに対して複数作成する。

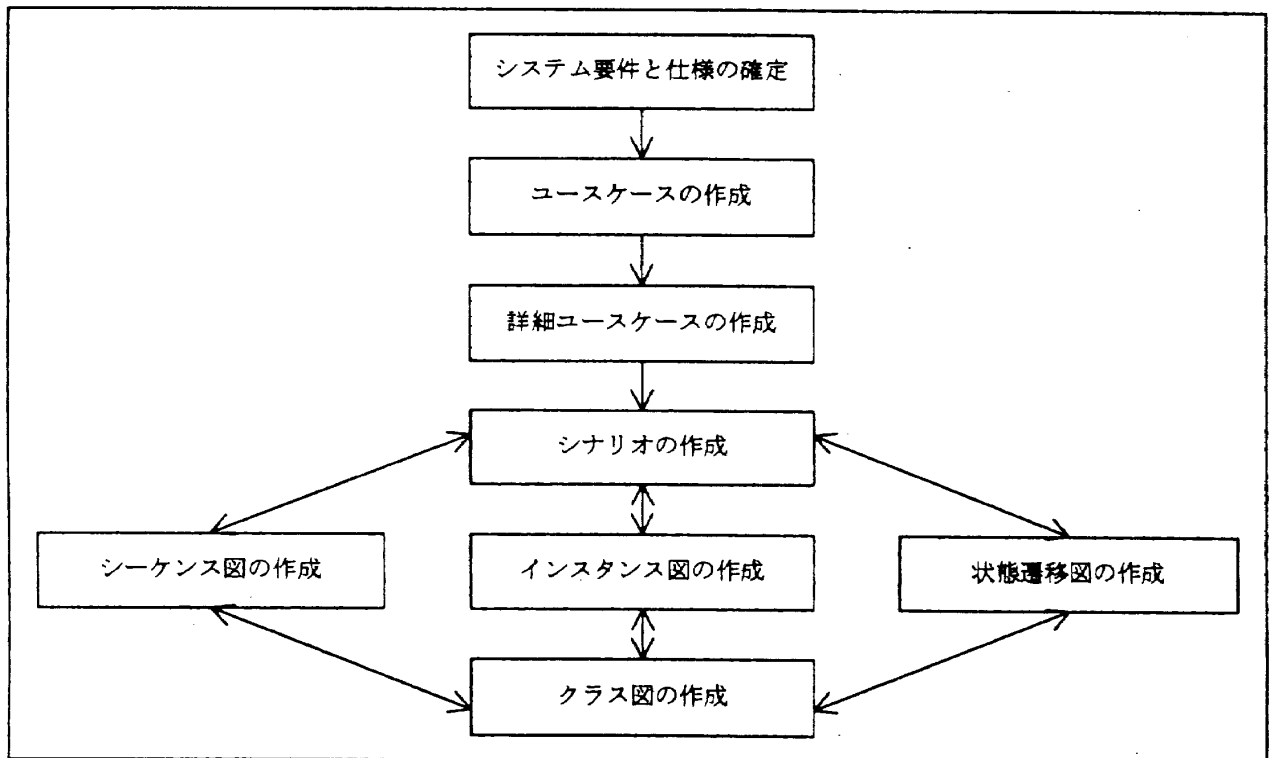
インスタンス図はオブジェクトから生成されるインスタンスを具体的に記述した動的モデルであり、オブジェクトの静的な関係と動的なシステムの流れが把握することが可能になる。結果的に、クラス図とシーケンス図の整合性確保が容易となる。

この「シナリオ」と「インスタンス図」を作成する工程を加えることで、分析・設計の作業工程のユースケースからのクラスの切り出しや、クラス図とシーケンス図の整合性確保が困難であるといった

ことが解決された。

二点目の作業行程の直列構成による、仕様変更に対する影響範囲への対応の工数が大きいという点については、従来の手法のように各工程が終了してから次の工程に進むのではなく、「ユースケース」・「シナリオ」・「インスタンス図」・「シーケンス図」・「クラス図」を相互に参照しながら作業を進めていくラウンドトリップ手法（図1-1）を適用することにより、仕様変更への対応・各ドキュメント間の整合性確保が容易になった。

図1-1 ラウンドトリップに構成した作業工程



以下に従来の分析設計手法と、前述した2点の問題を解決する手法と特徴を表にまとめた。

表1-1 分析設計手法の比較

	従来の分析設計手法	新分析設計手法
業務モデルの事象表現	詳細ユースケース	シナリオを作成
開発アプローチ型	ウォーターフォール型開発	ラウンドトリップ型開発
ドキュメント間の整合性	前後に作成されたドキュメント	全ドキュメント

このオブジェクト指向分析・設計における問題点の解決は、オブジェクト指向についての初心者においても、システム分析・設計の経験さえあれば、スムーズに対象業務の分析・業務システムモデルの設計を行うことのできる手法であると考えられる。