

UMLの要求仕様からZIPCを使用した 状態遷移表でのモデル検証成果

株式会社ネットタッチソフトウェア 開発部 主任技師

須藤 崇

膨大化および複雑化するSoC開発に対して、上流設計における開発プロセスの確立が急務ではないでしょうか。ここではUMLを使用した要求仕様分析からZIPCを使用した状態遷移表でのモデル検証を行なった成果について事例をご紹介します。

【背景】

実装完了後に問題点が発覚し、SW/HWともに修正に膨大な時間を費やしてしまうことは往々にしてあります。その原因として最も回避したい問題が、仕様および設計に関わるバグであると認識しています。この問題を回避するために、UMLを使用した要求分析を行い、その分析モデルをシミュレーションによる検証することにより、上流工程において仕様の洗練および詳細化を行うことで手戻りのない開発プロセスの確立を目指すことになりました。

ここではあるネットワークシステム開発（ハードウェア開発含む）で行なったプロセスに関してご報告いたします。

開発途中の業務であるためシステムに関する具体例などはご紹介出来ませんがご了承下さい。

【期待できる効果】

UMLを使用し、要求仕様をモデル化し検証することにより、以下のことが可能となるのではないかと期待を持ち作業を行ないました。

- ・システムを構成する機能を割出し、その妥当性の検証が可能となること。
- ・その機能同士の相互作用を分析することにより、管理すべき状態および振る舞いに関する検証を行なう事が可能となること。
- ・上記の検証結果を基に要求仕様が洗練（詳細化）されたものになること。

【要求モデル検証プロセス】

開発フローについて説明します。

今回は<図1>のような流れでモデリングを進めました。

- ①まず、「要求仕様（分析）」の工程では、ユースケースを記述しシステムが提供する機能の洗い出しを行ないます。
- ②次に、「ユースケース分析」の工程では洗い出した機能ごとの手順の詳細化を行ないました。
- ③続いて、「分析モデリング」の工程ではシステムを構成するクラスの抽出と振る舞いの検討およびクラス間の相互作用の検討を行ないます。
- ④最後に、「モデル検証・詳細化」の工程において上記工程にて導き出された分析モデルをシミュレーションにより検証し、補足および詳細化を行ないます。

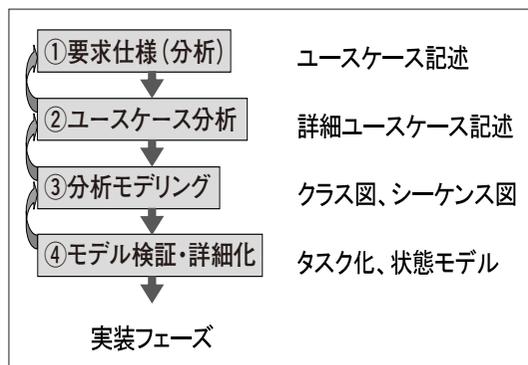


図1 開発フロー

<図2>に開発フローの全体像と各工程における成果物と各工程においてどの成果物を使用し分析を行なったかをまとめました。

では、各工程の作業を順番に説明していきます。

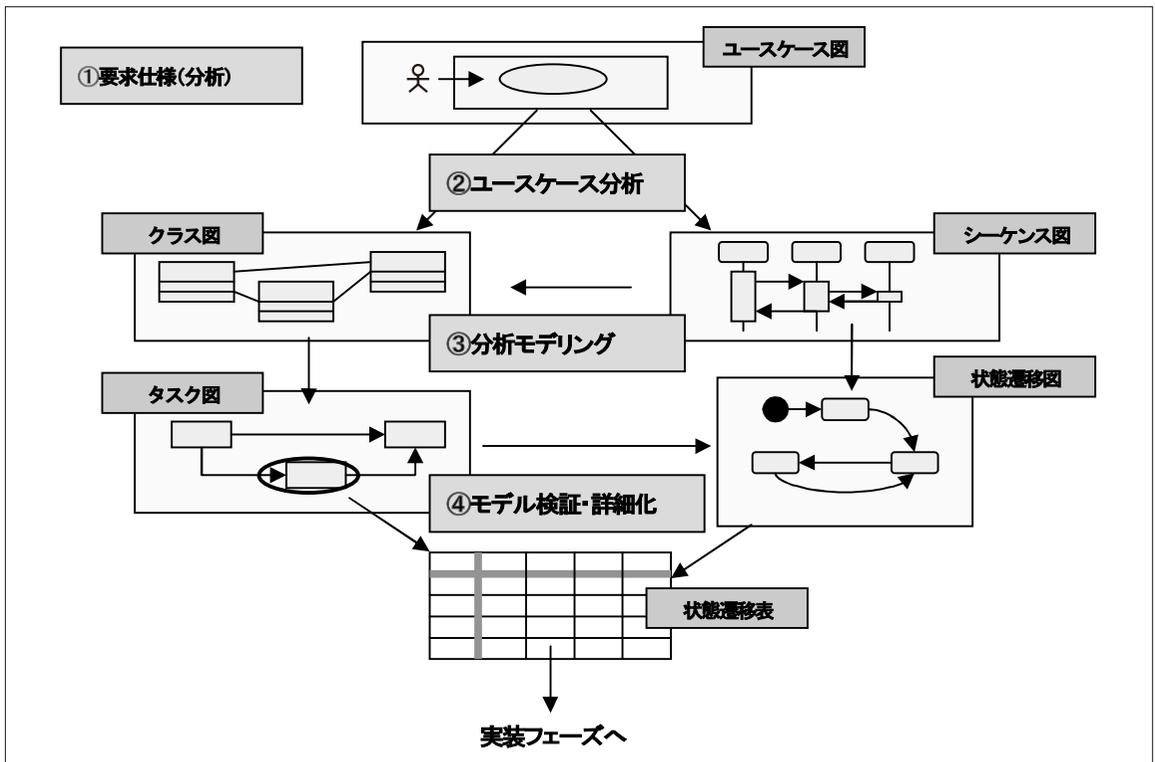


図2 分析・設計・検証のイメージ

まず最初に「要求仕様の分析」を行ないました。ここでの目的はシステムが提供する機能を明確にすることです。「ネットワークノードは何を行なうものか?」ということ洗い出す作業となりました。例えばあるネットワークノードでは「受信フレームからコマンドを抽出する」というように定義していきました。本工程の成果物は、ユースケースとユースケース図です。

次の工程は、「ユースケース分析」を行ないます。

この工程では前工程で抽出したユースケースをどのような手順で実現するかを検討します。ここではユースケースシナリオと概略シーケンス図を記述し、ユースケースを実現する際の具体的な手順および流れを定義していきました。本工程の成果物は、ユースケースシナリオと概略シーケンス図です。出来るだけ多くの例を記述し、後工程で行なうブラックボックステストに利用することが出来ました。

次の工程では、「分析モデリング(静的モデル)」を行ないます。

システムの構成要素である「クラス」の候補を洗い出し、責務に応じた振る舞いを検討する作業です。具体的にはクラスの抽出とクラス間の協調関係をクラス図を使用して検討していきます。前工程のユースケース、ユースケースシナリオを記述することによりすでに主要なクラス抽出は終わっていたというのが実感です。

次の工程は、「分析モデリング(動的モデル)」を行ないます。

実現すべき機能毎に、アクターおよび前工程で抽出した「クラス」間の相互作用の詳細を表現します。シーケンス図を記述し「クラス」間のメッセージの送信順、受信順が明確になりました。後工程で行なうホワイトボックステストにこの成果が利用出来ました。

最後の工程として、「モデル検証・詳細化」を行ないます。

分析モデルを検証することにより、要求仕様の詳細化することを目的としています。ここではシーケンス図を元に各クラスの状態遷移図(ステートチャート図)を記述しました。ZIPC

のコンバート機能を利用し、状態遷移図を状態遷移表 (STM) へコンバートし、シミュレーション機能を使用して各状態に対するイベント処理を検証します。

シミュレーションではユースケースの動作を確認するためのブラックボックステストとして各ノードに対して入力イベントを与えることにより、シミュレーション結果がユースケースと一致することを確認しました。また、ホワイトボックステストとして状態遷移表の全ての状態におけるイベント動作を確認することを行いません。

この結果、モデルの不備点などの明確化だけでなく要求仕様の矛盾点や詳細検討すべき点を明確にすることが出来ました。

【成果】

成果として、モデリングを行なうことにより以下の成果が得られました。

- ・ユースケースにより、提供するサービスを

構成する概念が整理することが出来た。

- ・上記概念から、必要な機能および各機能の関連を明確にすることが出来た。
- ・各機能間の相互作用をさらに分析することにより、アクティブタスクが抽出出来た。

シミュレーションを行なうことにより以下の成果が得られました。

- ・上流工程で通信ネットワークのフレーム中継伝達の方式・手段を検証出来た。
- ・検証結果をもとに要求仕様を詳細化し、実装可能とした。

【今後のフェーズ】

本プロジェクトの今後の予定として、今回のモデルを基にSW/HW分割を行い、XModelink SoC Modeler により、<図3>の様にSystemC構造図にてグラフィカルな設計からSystemCコードの自動生成を行い並行動作の検証確認、さらにはSystemCからVHDLへ変換を行いターゲットボードへの実装を行っていく予定です。

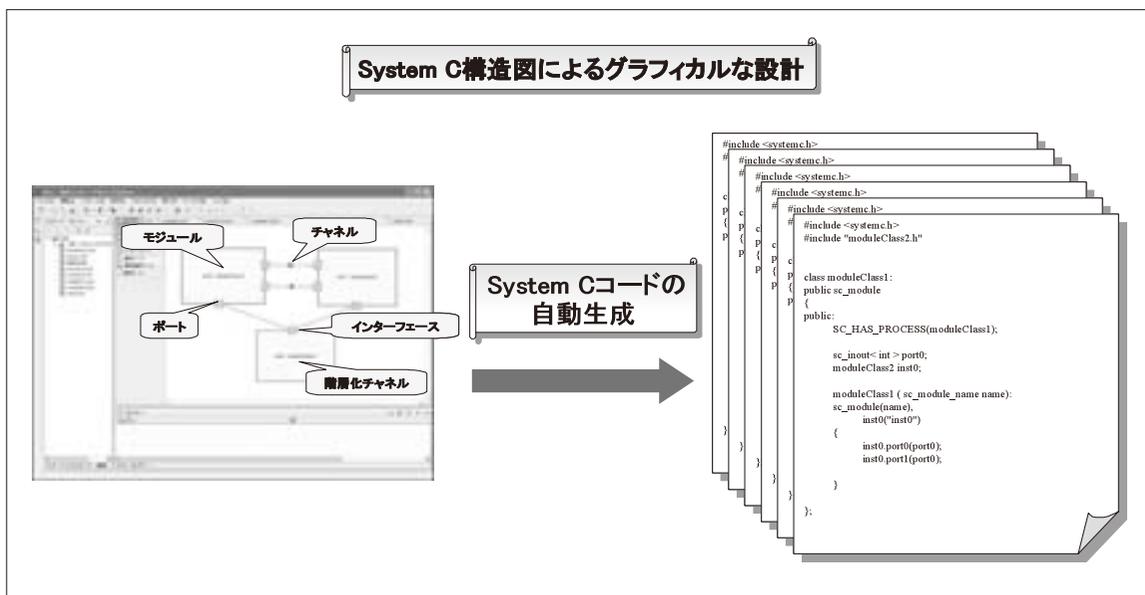


図3 XModelink SoC ModelerによるSystemC設計

