

リアルタイムシステムの性能評価・改善向け ドメインモデルの提案

田村恭久、伊藤 潔
上智大学 理工学部

ドメイン分析/モデリング技法を用いたリアルタイムシステムのドメインモデルを紹介する。リアルタイムシステムは処理すべき機能の仕様化や設計だけでなく、末端装置からの処理要求に対する応答時間の制約の仕様化や、時間制限を検証し、あるいは性能改善を行なう設計プロセスが必要である。一方、ドメイン分析/モデリングの技法を使って、似た性質を持つシステムのグループ（ドメイン）に独特な仕様モデル、解のモデルを開発し、これを再利用することで、実際のシステム開発の生産性を向上できる。著者らが提案しているTriadic Domain Model (TDM)を適用し、ドメイン固有のシステム開発プロセスを明かにすることにより、システムの生産性は更に向上できる。本稿では、リアルタイムシステムで用いられる技術であるデータベース、並列処理、分散処理を実装する際のプロセスから、リアルタイムシステムのドメインの設計には機能設計と性能設計の間を行き来する「行きつ戻りつプロセス」が特徴的であることを論じた。

Domain Model of Real-Time Systems for Performance Evaluation and Improvement

Yasuhisa Tamura, Kiyoshi Itoh
Faculty of Science and Technology, Sophia University
Kioi-cho 7-1, Chiyoda-ku Tokyo 102 JAPAN
{ytamura, itohkiyo}@sophia.ac.jp

This article introduces a domain model for real-time systems. Domain analysis / modeling (DAM) technique analyses a domain, which means many systems that have common target or solution, and builds a problem model and a solution model for this domain in order to reuse this model for newly developing systems. Our proposing Triadic Domain Model (TDM) analyses and clarify a model to describe a problem commonly to the domain (domain problem model), a model of a system solution commonly to the domain (domain Product model), and a model of a process to acquire the domain product model from the domain problem model (domain process model). As a result of applying this TDM technique to the real-time systems, two domain product models, functional one and performance one, become clear. To acquire this two product models, a real-time system designer would walk "go-and-back" process to make detail the functionality and to verify the performance. This process is characteristic to the real-time system domain.

1. はじめに

リアルタイムシステムとは、コンピュータが末端装置（顧客端末やプラントのセンサなど）から処理要求を受け付けてから末端に結果を返すまでの間の時間に制約があるシステムである。リアルタイムシステムの例としては、フライトシミュレータ、株の売買システム、化学や鉄鋼などのプロセス制御システムなどがある。株の売買システムでは、証券会社などのオペレータが売りや買いの注文を出す。一方、株価は絶えず変動している。オペレータの注文と実際の処理に数十分の時間差が生じると、オペレータが意図したものと異なる価格で株を取り引きする結果となるので、注文は遅滞なく処理される必要がある。どの程度の遅延時間まで許されるかは、対象とするシステムによって異なる。

リアルタイムシステムの仕様化や設計のプロセスでは、従来のソフトウェアにおける機能面の仕様化や設計だけでなく、性能面の仕様化や設計を同時に行なう必要がある。この両者は並行して行なわれ、かつ相互に関連を持ち、常に整合性を保つ必要がある。これに対し従来リアルタイムシステム向けに提案されてきた開発方法論の多く^{(2),(4)}は、ペトリネットなど並行処理系の記述や検証向けのツールを採用したり、あるいはGPSS、SIMSCRIPT などの性能評価ツールを応用している。これらの方法論は、性能の仕様化や検証には好適であるが、リアルタイムシステムの機能設計と性能設計をどのように関連づけ、整合性を保つかという問題に対する本質の解答になっていない。

本稿では、ドメイン分析／モデリング手法をリアルタイムシステムのドメインに適用し、機能と性能の仕様化・設計・検証を並行して進めるプロセスに好適なプロセスモデルを提案する。ドメイン分析・モデリング手法（Domain Analysis / Modeling : DAMと略す）とは、共通した対象問題や特徴を持つ複数のシステム、すなわちドメインに対して、システム開発に有効な共通の用語、問題の捉え方、システムの構造、システムの開発プロセスなど、対象ドメインに固有の概念構造を分析し、そのモデルを得るプロセスである。概念構造を分析するプロセスをドメイン分析、その結果得られた概念構造をドメインモデルと呼ぶ。例えば、A銀行・B銀行のオンライン処理など、解くべき問題が複数ある時、これらからオンライン処理問題のドメインを抽出できる。銀行には、顧客、口座、支店などの構成要素、また預金の預け入れや払い戻し、借り入れなどの動作、さらに口座番号や利率などの属性がある。これらオンライン処理に共通する概念が、オンライン処理問題の

ドメインを構成する。ソフトウェア開発においてDAMを導入することによって、ドメインモデルを用いて実際のシステム開発の生産性を向上し、再利用の促進することが期待できる。

本稿では、リアルタイムシステムのプロセスモデルを得るため、筆者らが提案しているTriadic Domain Model (TDM)⁽¹⁾を適用する。TDMとは、ドメイン分析の結果得られるドメインモデルを、次の3種類に分類するという考え方である。

(1) Domain Problem Model

…対象問題ドメインのモデル。問題の記述のみから成り、解法の記述は含まれない。問題ドメインの構成要素や語彙、データの型や数値の範囲、構成要素間の相互関係、個々の構成要素のダイナミクスなどの記述から成る。

(2) Domain Product Model

…解のドメインのモデル。解のためのデータ構造、アルゴリズム、資源、ライブラリなどの記述から成る。

(3) Domain Process Model

…Problem ModelからProduct Modelを導くプロセスのドメインのモデル。Product Modelで採用されるデータ構造、アルゴリズム、資源、ライブラリなどを取捨選択、あるいは新規開発、チューニングを行なうプロセスから成る。

これらのドメインモデルを用いたDAMとシステム開発のプロセスを図1に示す。DAMの元になるのは、対象問題がターゲットとなるドメインに属しているような過去の開発例である。DAMプロセスでは、これらを元に上の3種類のドメインモデルを開発する。これ以降のシステム開発では、問題の分析はDomain Problem Modelと対象問題の差分を求める作業となり、工数が低減される。またシステムの開発はDomain Process Modelで得られた開発プロセスに基づいて行なうので、後戻りや重複作業が低減される。このようにDAM実施後のシステム開発は、DAMで得られた様々なモデルを利用して行なえるので、DAM以前のそれと比べて一般的に効率が上がる。

TDMに基づいてドメイン分析を行なうことによって、ドメインモデルが分析者の立場、すなわち開発の依頼や設計の請け負いなどによって偏ることなく、どの立場からの分析やモデルかということが明確になる。また、問題や仕様から解を得るプロセスが、この両者のマッピングという位置付けによって明確になる。このような長所をもつことから、TDMはドメイン固有のプロセスモデルを開発するのに好適な手法であることがわかる。

リアルタイムシステムを特徴づける構成要素として、一般には並行プロセス処理、データベースアクセス、分散アーキテクチャなどが挙げられ

る。これらの各々は実現手段の1つであり、TDMの立場ではDomain Product Modelに属する。これに対するDomain Problem Modelに属する構成要素としては、個々の処理、扱うデータ、処理に対する制限時間などが挙げられる。よって、リアルタイムシステムドメインの Domain Process Model とは、要求仕様を満足する性能を得るための実現手段を選択し、またその性能を検証するプロセスであることがわかる。

Domain Product Modelに属する実現手段は、リアルタイムシステムを構成するための必須の構成要素ではなく、あくまでも時間制限を守るための手段である。それらのうちどれを選択するか、あるいは組み合わせるかは、設計者に任せられている。また、性能設計には性能評価・改善のプロセスが不可欠である。性能評価のモデルやパラメータは、機能の詳細や前述の実現手段が明確になると同時に詳細になる。よって、機能設計の進展と並行して性能評価が行なわれるプロセスモデルとなる。評価の結果所望の性能が得られないと判断した場合は、機能などを変更して性能を満たすようなフィードバックがかかる。このように機能設計と性能設計が並行して進展していく。この取捨選択や組み合わせ、性能の検証や性能改善といったプロセスは、リアルタイムシステムドメインのプロセスモデルに固有のものであると考える。

2. リアルタイムシステムのプロセスモデル

リアルタイムシステムが持つ特徴として、本稿では次の4点を取り上げる。

- (1) 処理に時間制限がある
- (2) データベースアクセス
- (3) 並行処理
- (4) 分散アーキテクチャ

このうち(2)~(4)は、リアルタイムシステムであるための必要条件ではないが、実際のリアルタイムシステムを観察すると(2)~(4)の技術を用いてシステムを実装している場合が多い。これらの実装技術は、設計フェーズにおいて明らかになるもので、要求仕様を確定する際には必ずしも採用が明確になっているとは限らない。よって、(2)~(4)はTDMではDomain Product Modelを実現するための技術である。

さて、リアルタイムシステムの構築に当たってDomain Problem Modelとして与えられるのは、扱うデータの概要と、当該データに対してなされる処理の概要、処理の制限時間である。データと処理だけを仕様として与えられても制限時間を守るよう実装できるか否かは判定できない。そこで上の(2)~(4)の技術の分析プロセスを利用する。

これらの技術を採用する場合のプロセスをまとめると次のようになる。

データベース

- ・データの正規化
- ・スキーマの一貫性チェック
- ・同時実行制御やロックの洗い出し

並行処理

- ・並列化可能なプロセス群の洗い出し
- ・排他制御／通信箇所の洗い出し
- ・複数処理間の共通事項の洗い出し
- ・デッドロック等の検証

分散処理

- ・分散可能なプロセス群の洗い出し
- ・データの一貫性を保証する機能の構築

例えば並行処理では、Problem Modelで与えられた処理の概要を分析し、並列に動かすことのできるプロセスを洗い出す。次に、それらのプロセス群が排他制御やプロセス間通信を必要とする箇所も洗い出す。また、複数プロセスで共通して処理できるものは共通化するので、該当箇所を洗い出す。更に、並行処理を行なうことによって生じる可能性のあるデッドロックなど不具合が生じるか否かを検証する。これらの作業を通して、動作が保証されている並行処理のフローがProduct Modelとしてできあがる。他にも、上に挙げたプロセスを用いて問題を分析していく。

分析のプロセスが進むと、Product Modelが持

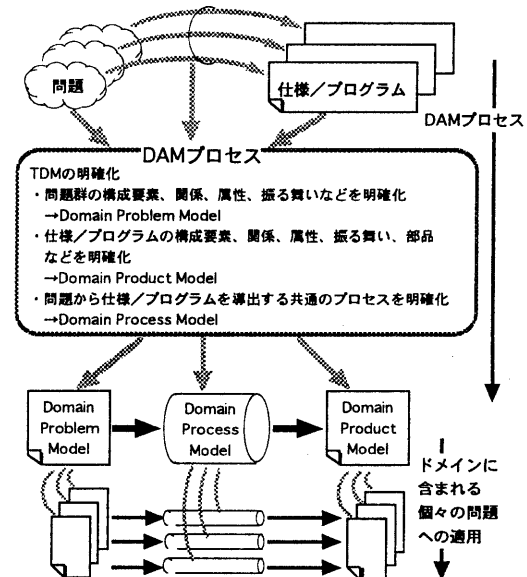


図1 Triadic Domain Model に基づくドメイン分析とシステム開発

つべき機能などが徐々に明かになっていく。また、処理の時間制限を検証するため、並列/分散処理フローを用いて待ち行列ネットワークを構築し、性能シミュレーションを行なう。この機能の詳細化と性能の検証は無関係ではない。すなわち、機能を明確にしていく一方で、その処理に要する時間を見積もり、この見積もりを処理フローに反映させる必要がある。機能を徐々に詳細にしていくプロセスのなかで、処理に要する時間の見積もりが変更になる場合もある。この場合は、その変更が処理時間にどのような影響を与えるのかを検証するため、迅速に性能モデルへフィードバックする必要がある。また、性能シミュレーションの結果処理が制限時間を超えてしまうことが判明した場合、処理フローを変える、ハードウェア変更などにより処理時間を短縮する、などの変更が必要になる。この変更も、性能モデルだけで閉じる問題ではなく、機能モデルへ影響を与える可能性があるため、迅速に機能モデルへフィードバックする必要がある。

このように、リアルタイムシステムの設計プロセスにおいては、機能モデルと性能モデルが相互に関連を持ちながら詳細化を進める。設計チームは機能モデルと性能モデルの間を行きつ戻りつしながら徐々に双方を詳細化していく。これがリアルタイムシステムの設計プロセスが持つ大きな特徴である。以上の議論をまとめTDMとしたものを図2に示す。

3. おわりに

本稿ではリアルタイムシステムのドメインが固

有に持つドメインモデルを取り上げた。リアルタイムシステムは機能面の設計と同時に性能面の設計を進めていく必要があり、設計者はこの両者の間を行きつ戻りつ設計を進めていくプロセスを辿る。よって、この「行きつ戻りつプロセス」がリアルタイムドメインに固有のプロセスモデルであることを論じた。

参考文献

- (1) 田村恭久、伊藤 潔、杵嶋修三、ドメイン分析・モデリング技術の現状と課題、情報処理 Vol.35, No.10, pp.952-961 (Oct. 1994).
- (2) Pyle,I., Hruschka,P., Lissandre,M., Jackson,K., Real-Time System : Investigating Industrial Practice, Wiley (1993).
- (3) Stankovic,J.A., Misconceptions About Real-Time Computing, Computer, Vol.21, No.10, pp.10-19 (Oct. 1988).
- (4) Calvez,J.P., Embedded Real-Time Systems, Wiley (1993).
- (5) Itoh,K., Kishima,S. and Tamura,Y., System Integration on Specification, Design and Generation of Reactive Systems - Triadic Domain Model-Based Approach, ICSI'94 (3rd Int'l Conf. on Systems Integration) (Aug.1994).
- (6) 伊藤 潔、田村恭久、杵嶋修三、Triadic Domain Modelに基づくシステムの分析・設計、FOSE'94 (Foundation of Software Engineering workshop) (Nov.1994).

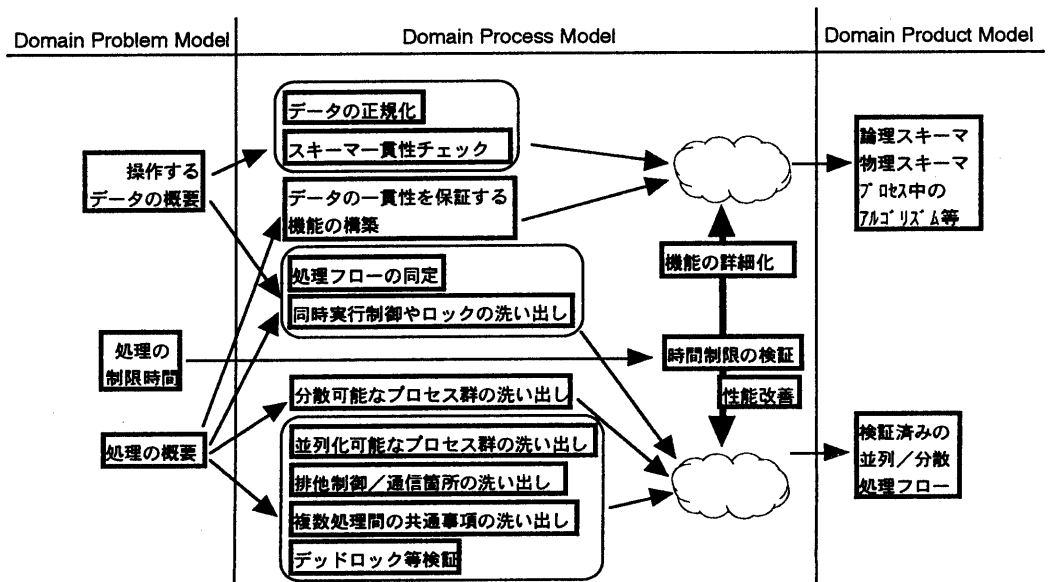


図2 リアルタイムシステムのTDM