5A-5

UML 要求分析モデルにおける CRUD 観点のデータライフサイクルの妥当性検査手法

青木　善貴[†]　　小形　真平[‡]　　奥田　博隆[†‡]　　松浦　佐江子[†‡]
日本ユニシス　株式会社[†]
芝浦工業大学　大学院工学研究科　機能制御システム専攻[‡]
芝浦工業大学　大学院理工学研究科　電気電子情報工学専攻[†‡]

１．はじめに

業務システム開発では，致命的な手戻りを回避するため，業務データのライフサイクルの妥当性を早期に保証すべきである．ここで，データのライフサイクルとは，ユーザとシステムのインタラクションにおいて，データが作成，読込，更新，削除（以降，CRUD と呼ぶ）される過程を表す．そして，妥当性のない定義とは，例えば「未作成・未取得のデータが更新される」といった実現不可能な定義を指す．

一般に，変更要求による仕様変更が生じやすい要求分析工程は，仕様中に矛盾が生じやすい．そのため，分析者が仕様を網羅的かつ効率的に検査できるように，モデル検査技術の早期の適用が期待されている．しかし，仕様が固定化しづらい要求分析への適用は困難である．

そこで，本稿は，これまで提案してきた UML(Unified Modeling Language)[1]要求分析モデルを対象に，データライフサイクルの妥当性を網羅的かつ効率的に検査できるように，モデル検査ツール UPPAAL[2]のモデルと検査式を完全に自動生成する方法を提案する．

２．UML 要求分析モデルと課題

２．１プロトタイプ生成可能な UML 要求分析モデル

我々はこれまで，業務系 Web システム開発を対象に，ユーザとシステムのインタラクションを表す要求分析モデルを顧客や開発者がわかりやすく理解できるように，ユーザの操作手順や入出力データを可視化するプロトタイプ（ユーザプロトタイプ）と，業務データの CRUD を中心としたシステム内の業務ロジックを可視化するプロトタイプ（システムプロトタイプ）が生成できる特徴を持つ，モデル駆動要求分析手法を提案してきた[3].

そして，要求分析モデルでは，インタラクションに着目し，正常／例外に渡る振舞いとデータのフローやデータの構造を UML アクティビティ図，クラス図，オブジェクト図を用いて表す．アクティビティ図では，振舞いやデータとこれらのフローをそれぞれ「アクション」，「オブジェクトノード」，「制御フロー」により表される．また，フローの分岐と結合は「ディシジョンノード」「マージノード」により表される．インタラクションの主体者であるユーザとシステムはそれぞれ「パーティション」により表される．なお，要求分析モデルの定義には，UML モデリングツールである astah*を用いる．

２．２データライフサイクルの妥当性保証の課題

顧客がユーザプロトタイプを通して，操作手順や入出力データを十分に確認し，分析者がシステムプロトタイプを通して業務データの CRUD の妥当性を吟味した場合でも，一度しか行わないデータの初期化等の実行頻度の低い振舞いを見落とすことにより，システム全体のフローとしては振舞いの順序が正しくなくなる．従ってシステム全体としてフローに矛盾がないことを網羅的かつ効率的に保証することは難しい．そこで，本稿では業務データの CRUD におけるライフサイクルに着目したモデル検査の自動化方法を提案する．

２．３モデル検査ツール UPPAAL

UPPAAL はグラフィカルなエディタとシミュレータ，検査器から構成される．エディタでシステムを構成するオートマトンを記述し，シミュレータによって分かり易く確認することが可能になっている．検査器ではデッドロックなどの性質や，オートマトンがある状態に到達するかのどうかの到達可能性のチェックができる．UPPAAL モデルは検査時にプロセスが割り当てられる．一つのモデルに複数のプロセスを割り当てて同時に実行させることもできる．

３．要求分析モデルの CRUD 観点の検査手法

本稿における提案は，分析者が要求分析モデル上のシステムパーティションに登場するエンティティクラスのライフサイクルオブジェクトノードの妥当性を確認するための，CRUD 観点の検査式と要求分析モデルから UPPAAL のモデルと検査式への変換方法である．図 1 に手法の全体像を示す．分析者は，まず業務データの CRUD 処理を明確化したアクティビティ図[4]を作成する．そして，UPPAAL の検査対象モデルをアクティビティ図から自動生成する．また，業務データを表すクラスやオブジェクトノードに CRUD 観点の検査式を適用し，UPPAAL の検査式を自動生成する．但し今回は，クラスごとにオブジェクトの CRUD を書くものとする．検査時に，検査対象モデルが検査式を満たさなかった場合，反例を頼りに要求分析モデルの修正箇所を特定し，分析者が手動で修正する．

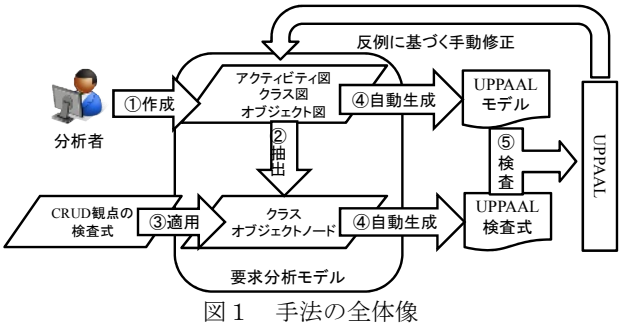

図１　手法の全体像

３．１CRUD 観点の振舞いとデータの関係記述

アクティビティ図中の単文の粒度まで分解されたアクションに対して，CRUD の意味を持つ動詞と目的語及び

Verification of UML Requirements Analysis Model from the Viewpoint of Data Lifecycle on CRUD
†Yoshitaka Aoki ‡Shinpei Ogata †‡Hirotaka Okuda †‡Saeko Matsuura
† Nihon Unisys, Ltd.
‡ Functional Control Systems, Graduate School of Engineering, Shibaura Institute of Technology
†‡ Electrical Engineering and Computer Science, Graduate School of Engineering, Shibaura Institute of Technology

それに合致したオブジェクトノードの位置関係を限定する．動詞は，適用事例で使用された動詞に基づき「作成する，検索する，更新する，削除する」等に限定した．

例えば，作成するアクションは，「○○な本を作成する」という形式で書かれ，目的語に作成されるオブジェクトノード名又はクラス名をとり，定められた動詞を取る．その結果として，そのアクションの直後に目的語に対応するオブジェクトノードを置く．

３．２UPPAALモデルの構成と生成

３．２．１UPPAALモデルの構成

検査モデルは 3 種類のモデルにより構成される．一つは処理フローのモデルである．処理の各ステップはシステムの状態遷移に基づきモデル化される．二つ目はオブジェクトのモデル(図 2)で，このモデルはオブジェクトの作成(Create)及びその前後の読込(Read)・更新(Update)・削除(Delete)の状態を持つ．図 2 上に検査条件になる各状態のロケーションを提示する．オブジェクト作成前の処理には先頭に”Pre”が付き，作成後の処理には”Post”が付く．処理フローのモデルと CRUD アクションの明示化により，1 対多のプロセス間で同期がとれる．三つ目はデータ作成判定を行うクラスのモデル(図 3)である．このモデルはオブジェクト初回作成と同期してロケーション Class に遷移する．以降変化しないためデータ作成の判断ができる．

３．２．２UPPAALモデル生成手順

処理フローのモデルは，astah*で作成したアクティビティ図から構成要素である「アクション」「ディシジョンノード」「マージノード」及び「制御フロー」を読み取る．要素ごとに定義された変換パターンに則って UPPAAL のロケーション及び遷移を作成する．オブジェクトのモデルとクラスのモデルは定形のものを使用する．

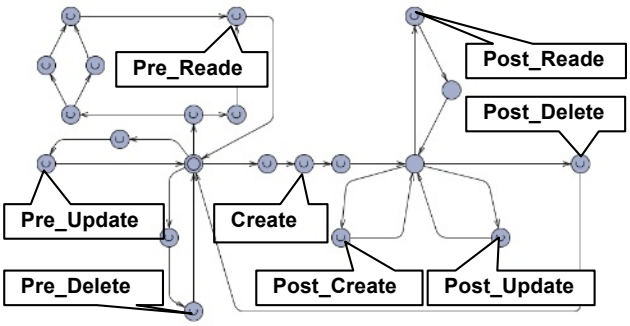

図２　オブジェクトのモデル

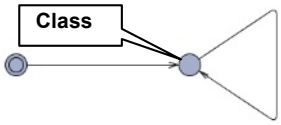

図３　クラスのモデル

３．３CRUD観点の検査式

本提案では，業務データのライフサイクルが実現可能であるように，次のように妥当性を定義する．

1) 未作成のデータが取得されてはならない
2) 更新/削除の前に作成/取得されていなければならない

これらを UPPAAL の検査式の形式である CTL (Computational Tree Logic)として表す．検査条件はロケーション(プロセス名．ロケーション名)を使用しており，到達時に true と判定される．

1) A[](not Cls.Class imply not Node.Pre_Read)
2) A[]((Node.Post_Update || Node.Post_Delete) imply (Node.Create || Obj_Node.Post_Read))

３．４反例によるモデルの修正

修正は手動により行う．反例の発生原因は，前節の妥当性の観点から振舞いの順序が適切でないことや，不要な振舞いが定義されていることの二通りが考えらえる．これらは，データが存在しない場合の例外フローや，アクションを削除することによって修正する．

４適用事例と考察

４．１教科書販売システム

本学大学院のソフトウェア工学の授業では，学内生協職員が教員のアンケートに基づき，学生への教科書販売を行うシステム（以下，教科書販売システム）の要求折衝を含めた要求分析を行う課題がある．課題は小規模なシステム構築の経験がある修士 1 年 3 人を 1 組で行った．

４．２考察

3.2 節で定義したオブジェクトのモデルを用いて上記課題のアクティビティ図を UPPAAL モデルへ変換し，想定シナリに沿って 3.3 節の検査式を用いたモデル検査を行なった．検査結果，新規授業に対してアンケートの作成ができないことが判明した．これは生協職員の処理が，アンケートの存在を前提としているためで，新規授業は，既存アンケートが存在しないため生協職員は処理を行えない．そして UPPAAL モデル上では未作成のデータの取得が行われることになり，オブジェクトのモデルのデッドロックが発生するので矛盾した呼び出しがされたとがわかる．他にも同様の実行頻度の低い振舞いの欠落を発見できた．処理が複雑になり作成した学生らでも把握しきれなくなったアクティビティ図から CRUD の妥当性を導き出すことが可能であることが確認できた．

５．関連研究

矢竹ら[5]は，複数のオブジェクトの協調動作の中で，振舞いが不変条件を満たすように状態を変化するかを，定理証明システムを用いて検証した．本手法では，データを厳密に検査できるが，そのデータに係わる振舞いの事前・事後条件を完全に定義する必要があるため，定義に手間がかかり容易な検査ができない．

本提案では，要求分析モデルでは厳密に定義されていないレベルであることから，検証できるレベルは違うが，システムが満たすべき性質の一つをこのように早期に，容易に検証できる．

６．まとめ

本稿では，ユーザとシステムのインタラクションを定義した UML 要求分析モデルを対象に，CRUD 観点のデータライフサイクルの妥当性を随時検査する方法を提案した．今後の課題として，本提案の導入の有無による欠陥発見の効率性の差異を評価する．

７．参考文献

[1]Unified Modeling Language, http://www.uml.org/
[2]UPPAAL, http://www.uppaal.com/
[3]小形真平，松浦佐江子，UML 要求分析モデルからの段階的な Web UI プロトタイプ自動生成手法，コンピュータソフトウェア，Vol.27，No.2，pp.14-32，2010.
[4]奥田博隆，小形真平，松浦佐江子，データライフサイクル保証の為の要求分析モデル検証手法の提案，日本ソフトウェア科学会第 28 回大会，2E-4，2011.
[5] 矢竹健朗，青木利晃，片山卓也，コラボレーションに基づくオブジェクト指向モデルの検証，コンピュータソフトウェア，Vol.22, No.1, pp.58-76, 2005.