5M-3

UMLクラス図のためのテキストベース言語および変換処理系

Text based language and conversion processing system for UML class diagrams

長谷部　篤史[†]　鷲崎　弘宜[††]　久保　淳人[†]　深澤　良彰[†]
Atsushi Hasebe　Hironori Washizaki　Atsuto Kubo　Yoshiaki Fukazawa

1　はじめに

Uniformed Modeling Language (UML)[1] は，ソフトウェア開発においてよく普及したモデル記法である。UML 図には，大きく 2 つの記述手段がある．両者の特徴を以下に示す．

手書き　**[長所]** 簡便で常に利用可能である．**[短所]** 作成された図の再利用が困難である．例えば，UML モデリングをモデリングツールに移行する場合，再度図を入力する必要がある．また，モデリングツールが提供する機能は利用不可能である．

モデリングツール　**[長所]** 記述の妥当性検査やソースコードへの変換といった高度な機能を利用可能である．**[短所]** 利用に先立って操作方法の習得の手間が必要である．また，計算機環境によっては利用不可能なときがある．

以下，UML モデリングツールを単にモデリングツールと表記する．

本稿では，UML を用いたモデリング作業として，

- 外出時など，高機能なモデリングツールが利用不可能な状況では設計アイディアを簡単に記述しておき，後にモデリングツールを用いて設計を発展させる場合
- メール，Wiki 等による開発者同士の対話的モデリング

を想定する．

手書きで UML 図を記述すると図の再利用が困難になるので，電子的に UML 図を記述する必要がある．PDA 等の携帯端末でも利用可能な記述手段として，テキストエディタが考えられる．また，複数人での共同作業では，成果物の可搬性が重要であるので，テキストを基礎とした記述形式が必要である．MOF に従ったモデルを XML 文書に変換する枠組みである XML Metadata Interchange (XMI) [2] は，テキストを基礎とした記述形式であるが，モデリングツール間のデータ連携を目的としているため，可読性が低く，人手による記述は困難である．したがって，(A) 可読性が高く，(B) 人手による記述が容易で，かつ，(C) モデリングツールでの作業に容易に移行可能な，新しい記述形式が必要である．

本稿において，我々は

1. 簡易かつ図式的に表現するテキストベース言語の定義 (上記 (A)(B) を満たす)
2. テキストベース言語による記述から XMI データへの変換処理系 (上記 (C) を満たす)

を提案する．提案テキストベース UML 記述言語は，クラス図を対象とする．

提案言語によって，記述手段が制限される状況でのモデリング作業から，モデリングツールを用いた本格的なモデリング作業への漸次的な移行が可能になる．また，テキストベースであることから，複数人の共同作業によるモデリングにおいて，モデリング成果物の可搬性の向上が期待される．

[†] 早稲田大学大学院理工学研究科
[††] 国立情報学研究所

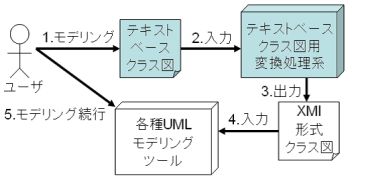

図1　Te-UML の環境を利用したクラス図モデリングの流れ

```
==============================
   & Component
------------------------------
------------------------------
 + operation() : void
 + addChild(component: Component) : void
 + getChild(index: int) : Component
==============================
==============================
    Composite
------------------------------
 - nodes : ArrayList
------------------------------
 + operation() : void
 + addChild(component: Component) : void
 + getChild(index: int) : Component
==============================
==============================
    Leaf
------------------------------
------------------------------
 + operation() : void
==============================
Component [child] ( 0..* ) <---------<> [parent] Composite
Component <|---------- Composite
Component <|---------- Leaf
```

図2　Composite パターンの Te-UML クラス図

2　テキストベース言語および処理系の提案

我々は，クラス図用のテキストベース言語である U 言語 [3] を拡張し，その言語に基づいた図を XMI 形式データに変換する処理系を実装した．我々はこの言語を「Te-UML」と呼ぶ．

UML クラス図のためのテキストベース言語を定義することにより，テキストの記述/閲覧可能な計算機環境上で、クラス図の記述を実現する．さらに，メールや Wiki を用いた複数人での対話モデリングにも適している．しかし，テキストで表現されるクラス図において，クラス図の全ての表記を対応するのは難しく，クラス図の詳細な内容の記述に適さない．Te-UML 言語は設計初期段階のクラス図の記述に適する．複雑なモデリングを行う場合は変換処理系により，記述した Te-UML クラス図を XMI 形式データへ変換し，モデリングをモデリングツールに引継ぐ．Te-UML 図を記述し，モデリングをモデリングツールに引き継ぐ流れを図 1 に示す．

Te-UML 言語はクラス図の表記全てには対応しないが，プログラムへの変換にあたり考慮すべき表記に対応する．クラスへの対応表記は，クラス，インターフェース，メソッド，プロパティ，static 等全ての修飾子である．関連への対応表記は，関連，汎化，依存，洗練，誘導可能性，集約，コンポジット，多重度，ロールである．

Te-UML 言語における大きな特徴は，クラス図と関連を分けて記述することにある．記述例として，Te-UML 言語で記述された Composite パターン [4] を図 2 に示す．

クラスは”=”の境界線ではさむことにより定義される．さらに，それら境界線の間に挿入可能な”-”の境界線により区画を

```
+<UML:Class xmi.id="Leaf" name="Leaf" version=
+<UML:Class xmi.id="Composite" name="Composite"
+<UML:Class xmi.id="Component" name="Component"
+<UML:Generalization xmi.id="Leaf-generalized-
+<UML:Generalization xmi.id="Composite-general
+<UML:Association xmi.id="0-Composite-Component
 <UML:AssociationEnd xmi.idref="relation-1-Com
 <UML:AssociationEnd xmi.idref="relation-1-Com
```

図3　変換された XMI 形式データの一部

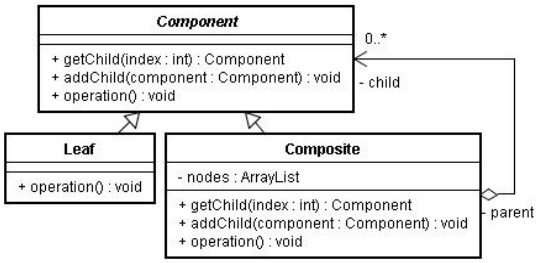

図4　モデリングツール「JUDE」へ出力した図

作り，上の区画から順にクラス名，プロパティ，メソッドを記述する．関連は，関連する二つのクラス名と，その間に各関連の意味を持つ線および矢印を記述して表現する．

変換処理系は Te-UML 言語に基づいた図を読込み，XMI 形式データを出力する．変換処理系を用いて，図 2 に示した Composite パターンの Te-UML 図から変換された XMI 形式データの一部を図 3 に示す．さらに，XMI 形式データがモデリングツールへインポートされた図を図 4 に示す．

変換処理系は JavaCC により実装され，Java 実行可能環境下により動作するため，利用可能な環境は広い．

3　評価

本研究と同様の目的を持つ U 言語，Silvertejp[6]，および一般のモデリングツールとの比較を行い Te-UML を評価する．U 言語には，Java への変換処理系 [5] が用意される．Silvertejp もクラス図用のテキストベース言語であり，同様の処理系が用意される．比較項目と評価を表 1 に示す．

表 1　Te-UML と関連研究およびモデリングツールとの比較

評価項目	Te-UML	U 言語	Silvertejp	モデリングツール
対応ダイアグラム	△	△	△	○
クラス図網羅	△	×	△	○
プラットフォーム	○	○	○	△
記述容易性	△	△	×	○
図の見やすさ	×	×	△	○
ファイルサイズ	○	○	○	△
XMI 出力	○	×	×	○

対応ダイアグラムにおいて，どのダイアグラムを対象とするか評価する．各言語はクラス図のみを対象とし，モデリングツールは UML の全ダイアグラムを対象とする．

クラス図網羅において，クラス図の表記にどの程度対応するかを評価する．Te-UML は主な表記を網羅し,Silvertejp も同様である．しかし U 言語は，ロールと，プロパティおよびメソッドの static 等可視性以外の修飾子に対応しない．モデリングツールはクラス図の表記を完全に網羅する．

プラットフォームにおいて，どのような OS の種類，ハードウェア性能の上で，クラス図の記述および変換処理系の実行が可能かを評価する．各言語はテキストエディタで記述可能であり，変換処理系は Java 実行可能環境下で動作する．モデリングツールには，Windows 系のみの対応で，または PDA 等携帯端末程度の計算機性能では起動不可能であるものが多い．

記述容易性において，図を表現するための記述量を評価する．Te-UML と U 言語では，記述量の少ない簡易な図を記述し，Silvertejp では，見やすいが記述量の多い図を記述する．モデリングツールは GUI での容易なモデリングが可能である．

表 2　記述されたデザインパターンの平均ファイルサイズ (byte)

Te-UML	U 言語	Silvertejp	モデリングツール
1347	1370	1538	3823

図の見やすさにおいて，モデリングツールで示されるようなクラス図にどれだけ近い図を記述可能であるかを評価する．Te-UML と U 言語では，クラスと関連を分けて記述するため不明瞭な図となり，Silvertejp ではクラスと関連を一緒にした図を記述可能であるため見やすい．

ファイルサイズにおいて，記述されたクラス図のサイズの大きさを評価する．各言語およびモデリングツールで記述された全 23 のデザインパターンの平均ファイルサイズを表 2 に示す．各言語間ではサイズに差はなく，モデリングツールから出力されるファイルのサイズは，他と比べて約 3 倍となる．

XMI 出力において，記述されたクラス図から XMI 形式データを出力可能か評価する．Te-UML では可能であり，U 言語，Silvertejp は不可能である．

以上の比較から考察を述べる．

テキストベース言語の比較では，Te-UML と U 言語では図を記述しやすく，Silvertejp では図が見やすいという点が異なる．テキストエディタはクラスや関連の少ない簡易な図の記述に適するため，見やすさよりも記述しやすさを強みとする Te-UML と U 言語の方がクラス図のためのテキストベース言語に相応しい．さらに，Te-UML はより多くのクラス図表記に対応する点で U 言語より優れる．

変換処理系の比較では，Te-UML では XMI 形式データを出力し，U 言語，Slivertejp では Java ソースを出力する点で異なる．Te-UML は，記述した図から XMI 形式データを出力し，モデリングツールを用いたモデリングの継続を実現する．

Te-UML とモデリングツールの比較では，Te-UML は OS や計算機性能に制限されない利用環境の広い点が際立つ．

4　おわりに

我々は，UML クラス図のためのテキストベース言語と，その言語に基づいた図の XMI 形式のデータを出力する変換処理系を実現した．UML は様々な視点から分析する図を用意するため，今後はクラス図以外の図にも対応する必要がある．

参考文献

[1] Object Management Group, Unified Modeling Language (UML), version 2.0, http://www.omg.org/technology/documents/formal/uml.htm

[2] Object Management Group, MOF 2.0/XMI Mapping Specification, v2.1, http://www.omg.org/technology/documents/formal/xmi.htm

[3] 平鍋健児, U 言語 [U-Language], http://www.objectclub.jp/technicaldoc/uml/u_lang/

[4] Gamma, e., et al.: Design Patterns: Elements of Reusable Object-Oriented Software,　AddisonWesley(1994).

[5] souceforge.jp, U Language Parser, http://sourceforge.jp/projects/ulparser/

[6] COLLABNET, Silvertejp, http://silvertejp.tigris.org/