モデル仕様の変更に適応可能なソースコード生成ツールの自動生成系の提案

M2008MM005 紅谷陽介
指導教員：野呂昌満

1 はじめに

近年，モデルからプログラムコードの自動生成をおこなう研究がされている．代表的な研究としてモデル駆動型アーキテクチャ(MDA)[2] がある．MDA の仕組みを図 1 に示す.

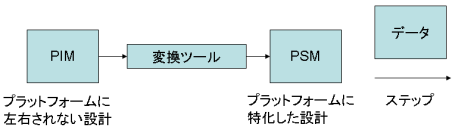

図 1 MDA の仕組み

MDA を利用したモデルからプログラムコードの自動生成において，モデルの仕様に対しての変更があった場合において，モデルの要素の情報から PIM(Platform Independent Model) への変換ツールに対して修正を加えなければならない．モデルの使用に対して変更があった場合，モデルとプログラムコードとの対応関係について修正をおこなう必要がある。モデルからプログラムコードへの生成ツールを作成するには修正した対応関係をもとにコーディングする必要があるので、モデル仕様の変更に対して柔軟性がないといえる。対応関係を関連する情報について整理をおこなうことで機械的に処理する事が可能である．本研究の目的は，モデルの仕様の変更に適応可能なソースコード生成ツールの自動生成系の提案をおこなうことである．本研究のアプローチとして，コード生成ツールにおいて必要な情報について整理をおこなう．次に整理された情報と言語要素との対応付けをおこなうことで，内部フォーマットの決定をおこなう．内部処理方式として言語処理技術の概念を適用する．提案する自動生成系は本研究で整理したコード生成ツールに必要な情報を入力とし，コード生成ツールを出力する．自動生成系で適用するモデルの事例として，本研究室で提案されている，ソフトウェアのアスペクト指向ソフトウェアアーキテクチャスタイル (Aspect Oriented Software Architecture Style for Embedded System :以後, E-AoSAS++)[1] のモデルを利用する．提案した自動生成系に対して妥当性，有用性に関する考察をおこなった.

2 E-AoSAS++

E-AoSAS++は組込みシステムのソフトウェアアーキテクチャを構築するためのコンポーネント・コネクタスタイルに基づく枠組であり，アプリケーション論理を実装するオブジェクトと，その状態遷移アスペクトを実現する並行状態遷移機械 (CSTM) を基本的な構成要素とする． E-AoSAS++では静的構造図としてコンポーネント図，クラス図．動的構造としてシーケンス図，状態遷移図を用いる．本研究では E-AoSAS++のモデルを記述するための UML モデリングツールとして EA(Enterprise Architect) を用いる．

3 背景技術

3.1 MDA

MDA は，モデルとして表現される構造的仕様のガイドラインを提供するソフトウェア開発手法である．MDA ではプラットフォームに独立モデルである PIM の設計を行い，プラットフォームに依存したモデルである PSM への変換を行うことで，複数のプラットフォームに対応したプログラムコードの自動生成が可能となる．

3.2 XMI,XMIDTD

XMI(XML Metadata Interchange)[3] は XML (Extensible Markup Language) を使ってメタデータ情報を交換する標準規格である．XMI の典型的な利用法として，UML モデルの交換形式としての利用がある．XMI を採用するメリットは，異なるメーカーの開発ツール間で，記述した設計モデルの互換性を保証する点である．ソースコード生成ツールの入力のデータ形式として XMI の利用を想定する．XMIDTD は XMI の文章構造情報を定義するためのスキーマ言語の一つである．XMIDTD では XMI の文章構造が厳密に定義されている．本研究では XMIDTD に付加情報を与えたものと，プログラムの仕様を入力として，XMI の字句解析器と構文解析器の出力をおこなうツールの自動生成系を提案する．

4 自動生成系の設計

4.1 自動生成系に必要な情報

本研究で提案する自動生成系を作成するために必要な情報は以下の５つであると考えられる．

- モデルの要素の情報
- モデル要素の情報が記述された文章構造の情報
- 記述されたモデルの意味情報
- モデルとプログラムコードとの対応関係
- プログラムコードの記述方法

本研究では E-AoSAS++のモデルを利用する．E-AoSAS++のモデルを UML ツールの１つである EA を使い作成する．EA からは XMI，XMIDTD を出力することができる．また，モデルからプログラムコードへの変換規則についての研究 [4] からプログラムコードの仕様を生成する．XMIDTD に意味情報を付加したものと，プログラムコードの仕様を入力とし，XMI からプログラ

ムコードの生成をおこなうツールを出力する．自動生成ツールに必要な情報と本研究で利用する要素との関係を以下に示す．

- モデルの要素の情報：XMI 記述
- モデル要素の情報が記述された文章構造の情報：XMIDTD
- モデルの意味情報：拡張 XMIDTD
- モデルとプログラムコードとの対応関係を表した情報：拡張 XMIDTD
- プログラムコードの記述方法：プログラムの仕様

XMIDTD を拡張するために必要な情報は，モデルの要素からプログラムコードへの対応情報を持つ．

4.2 自動生成系の処理方法

本研究における自動生成系の概要図を図 2 に示す．

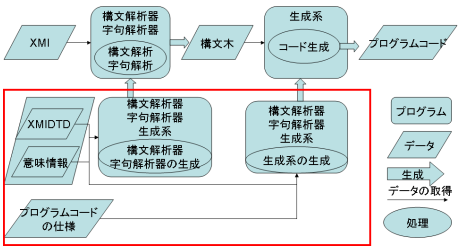

図 2 提案する自動生成ツールの概要図

モデル要素の情報が記述された文章構造の情報と，モデルの意味情報を持つ XMIDTD を入力とし，言語処理系に通すことでモデルの要素の情報を表した XMI 記述を解析するために必要な構文解析器と字句解析器を出力する．出力された構文解析器，字句解析器は XMI 記述を解析し，構文木を作成することができる．モデルとプログラムコードとの対応関係の情報を持つ拡張 XMIDTD とプログラムコードの仕様から構文木を入力とし，言語処理系を用いてプログラムコードの出力をおこなう生成系を出力する．モデル要素の情報が記述された文章構造の情報，モデルの意味情報，モデルとプログラムコードとの対応関係を表した情報，プログラムコードの記述方法から構文解析器，字句解析器，生成系を出力することができれば，モデル仕様の変更に適応可能なソースコード作成ツールの自動生成をおこなうことができる．対応関係の情報を自動生成系で利用する事で，モデルの意味の変更に対応することが可能となる．モデルの要素の情報が記述された文章構造の情報を利用する事でモデルの記述方法の変更に対しても対応可能となる．

4.3 アプローチ

コード生成ツールにおいて必要な情報について整理をおこなう．次に整理された情報と言語要素との対応付けをおこなうことで，内部フォーマットの決定をおこなう．内部処理方式として言語処理技術の概念を適用する．提案する自動生成系は本研究で整理したコード生成ツールに必要な情報を入力とし，コード生成ツールを出力する．

- モデル仕様の変換に対してソースコード生成ツールをおこなうために必要な情報について整理する
- ソースコード生成ツールを自動生成するための技術として言語処理技術を用いる．字句解析，構文解析さえできていればアクション部分に対して変更をおこなうだけでよい．特にモデル要素の情報が記述された文章構造の情報に対して変更を加える可能性が高いので柔軟性と拡張性を持つ

4.4 E-AoSAS++による図式情報と取得する情報

本研究では E-AoSAS++における自動販売機の静的構造図を例に説明する．図 3 に自動販売機の静的構造図を示す．

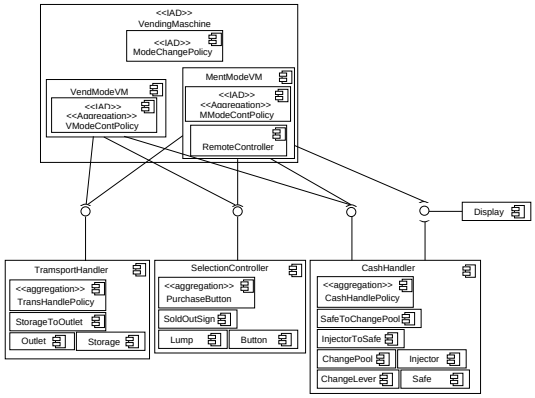

図 3 自動販売機の静的構造図

図 3 に示す静的構造図から，本研究で提案する自動生成系は状態遷移機械名を取得する．図 3 に示す静的構造で示されるコンポーネントは状態遷移機械を表現しており，各コンポーネントのコンポーネント名を取得することで，状態遷移機械名を取得する．

4.5 E-AoSAS++によるモデルから XMI 記述と XMIDTD の生成

図 3 で表した静的構造図から XMI 記述の出力をおこなった．出力した XMI 記述を図 4 に示す．

```
- <UML:Component name="VModeContPolicy" xmi.id="EAID_C639DBBC_D346_4323
    namespace="EAPK_6BF277A8_60D7_4bc2_AE92_633798A505B6" isRoot="fals
  - <UML:ModelElement.stereotype>
      <UML:Stereotype name="IAD" />
    </UML:ModelElement.stereotype>
  - <UML:ModelElement.taggedValue>
      <UML:TaggedValue tag="isSpecification" value="false" />
      <UML:TaggedValue tag="ea_stype" value="Component" />
      <UML:TaggedValue tag="ea_ntype" value="0" />
      <UML:TaggedValue tag="version" value="1.0" />
      <UML:TaggedValue tag="package" value="EAPK_6BF277A8_60D7_4bc2_AE92
      <UML:TaggedValue tag="date_created" value="2010-01-29 01:04:35" />
      <UML:TaggedValue tag="date_modified" value="2010-01-29 01:04:59" />
      <UML:TaggedValue tag="gentype" value="Java" />
      <UML:TaggedValue tag="tagged" value="0" />
```

図 4 自動販売機の静的構造図の XMI 記述

図 4 は自動販売機の例の VModeContPolicy を表している．次に出力した XMIDTD について図 5 に示す．

XMIDTD では ELEMENT のあとの XMI のタグに構文情報が書かれている．ATTLIST のあとにはモデルの要素に対しての詳細表現がある．

```
<!ELEMENT UML:Component (UML:ModelElement.name |
                         UML:ModelElement.visibility |
                         UML:GeneralizableElement.isRoot |
                         UML:GeneralizableElement.isLeaf |
                         UML:GeneralizableElement.isAbstract |
                         ・・・・・・・・・・・・・・・)*
<!ATTLIST UML:Component
          name CDATA #IMPLIED
          visibility (public | protected | private) #IMPLIED
          isRoot (false | true) #IMPLIED
          ・・・・・・・・・・・・・・>
```

図 5　自動販売機の静的構造図の XMIDTD

4.6　XMIDTD から XMI 構文解析器への変換規則

XMIDTD と XMI 構文解析器の変換規則の定義をおこなった．XMIDTD から構文解析器への変換規則を図 6 以下に示す．

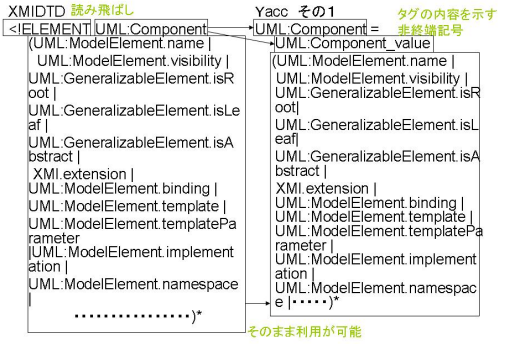

図 6　XMI の図情報を入力とする構文解析器

ELEMENT という情報があった時には XMI のタグを表すことがわかる．ELEMENT の次にくるタグを非終端記号とし，タグの内容を示す非終端記号を新たに作成する．

4.7　XMIDTD から XMI 字句解析器への変換規則

次に XMIDTD から字句解析器への変換規則を図 7 に示す．

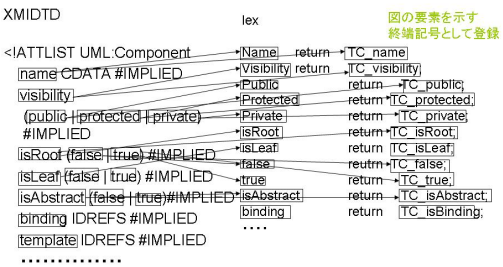

図 7　XMI の図情報を入力とする字句解析器

XMIDTD から字句解析器を出力する方法について記述した．読み込む値として XMIDTD の名前を入れる．また終端記号として TC1_+名前として登録する．

4.8　構文解析器によって作成される構文解析木

図 3 の静的構造図の XMI 記述から作成する構文木を以下に示す．

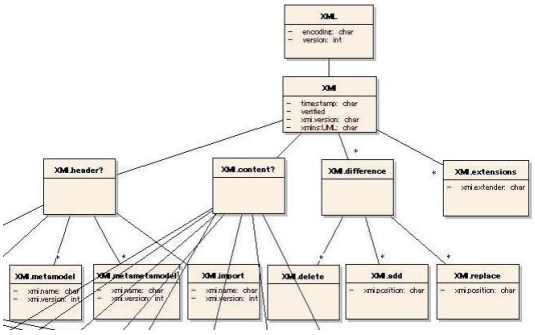

図 8　自動販売機の静的構造図の XMI の構文木

作成する構文木のモデルを EA のクラス図を用いて表現した．クラスにはタグの部分が対応し，属性の値にはタグの中の値が対応する．

4.9　構文解析器におけるアクション部分の記述

構文解析器のアクション部分についてモデルの要素から対応関係を定義した．対応関係を図 9 に示す．

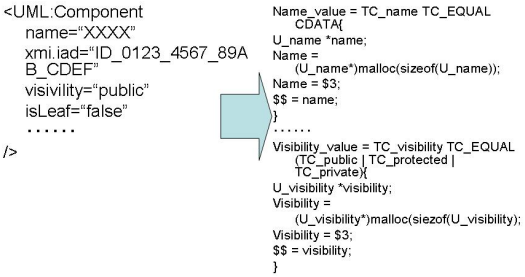

図 9　図の情報とアクション部分との対応関係

モデル要素の構文と構文木を作成するコードを表している．XMIDTD から XMI の構文解析器，および字句解析器を生成することができることがわかった．モデル要素の情報が記述された文章構造の情報を与える事でモデルの要素の情報の構文解析器，および字句解析器を自動生成することができる．

5　自動生成系における意味情報の付加と生成系

生成系の自動生成系では拡張 XMIDTD とプログラムコードの仕様を入力とし，XMI の生成系を自動的に生成する．

5.1　拡張 XMIDTD

拡張 XMIDTD の付与情報について図 10 に示す．

モデルの要素の右側にモデルの要素の意味を定義した言語への表示的情報の定義をおこなう．この定義ではモデルの要素とモデルの要素の意味を定義した言語への 1 対 1 の対応を表現することができる．最下部にモデルの要素の意味を定義した言語からプログラムコードへの表

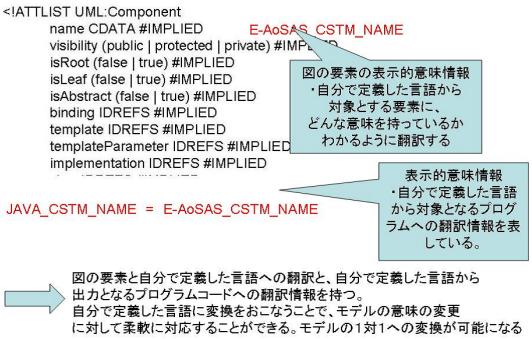

図 10　拡張 XMIDTD

示的意味情報の定義をおこなう．この定義ではモデルの要素の意味を定義した言語からプログラムコードへの 1 対 1 の対応を表現することができる．

5.2　プログラムコードの仕様の例

E-AoSAS++のモデルから Java プログラムコードへの変換論理に関する研究がされている [4].

本研究ではプログラム中において定型コードでない部分に識別子を与えることで，プログラムコードの仕様を作成した．プログラムコードの仕様を図 11 に示す.

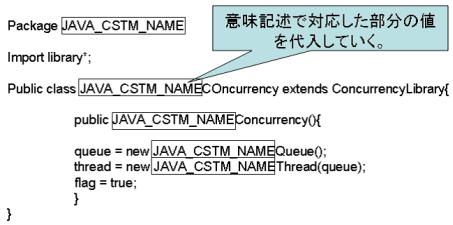

図 11　出力するプログラムコードの仕様

拡張 XMIDTD を入力として構文木からプログラムコードへの変換をおこなった．モデルの意味情報と，モデルとプログラムコードとの対応関係を表した情報，プログラムの仕様を与える事で，構文木からプログラムコードの生成系の自動生成をおこなうことができる.

6　考察

本研究で提案したソースコード生成ツールの自動生成系を以下の 3 つの観点から考察することで，提案した自動生成系に関する考察をおこなう.

- 内部フォーマットと内部処理方式の妥当性
- モデルとプログラムコードとの対応関係の拡張可能性
- 提案した自動生成系についての有用性

6.1　内部フォーマットと内部処理方式の妥当性

提案したソースコード生成ツールの自動生成系では，モデルの仕様の変更があった場合，モデルの要素の記述方法とモデルの意味情報を変える事で変更をおこなうことが可能である．モデルの要素の記述方法はツールによって出力可能であり，モデルの意味情報はモデルとプログラムコードとの対応関係と切り分けられており，修正が容易であると考える．また，出力するプログラムコードに対しての変換があった場合においては，モデルの意味から出力するプログラムコードとの対応関係について整理する事で変更をおこなうことができる．モデルの意味から出力するプログラムコードとの対応関係はモデルの記述方法から切り分けられており，修正が容易であると考える.

6.2　モデルとプログラムコードとの対応関係の拡張可能性

本研究で提案した自動生成系ではモデルとプログラムコードとの対応関係を 1 対 1 であるものとした．対応関係の記述方法を変える事でモデルとプログラムコードとの対応関係の拡張をおこなうことが可能であると考える．モデルの要素の一部の利用，モデルの要素に追加した情報を利用することが可能になる.

6.3　提案した自動生成系についての有用性

本研究では E-AoSAS++から Java 言語に変換するツールの自動生成をおこなうことができた．E-AoSAS++の仕様変更がおこなわれた場合には，モデル要素の情報が記述された文章構造の情報とモデルの意味情報に対して変更をくわえる必要がある．モデル要素の情報が記述された文章構造の情報は EA によって出力することが可能である．モデルの意味情報の変更をおこなうことで仕様変更に対しても柔軟に対応する事ができる.

7　おわりに

本研究では，モデルからプログラムコードの自動生成における問題点について着目した．モデルの仕様の変更に対して柔軟性がなく，対応関係について修正をおこない、対応関係をもとに修正をおこなわなう必要がある。自動生成に必要である情報について整理をおこない，ソースコード生成ツールの自動生成系の提案をおこなった．モデル要素の情報が記述された文章構造の情報，モデルの意味情報，モデルとプログラムコードとの対応関係，プログラムコードの仕様の情報を与える事でモデルからソースコード生成ツールの自動生成をおこなうことができる.

参考文献

[1] M.Noro, A.Sawada, Y.Hachisu, and M.Banno “E-AoSAS++ and its Software Development Environment”, *Proceedings of the 14th Asia-Pacific Software Engineering Conference(APSEC2007)*, pp. 206-213, Dec. 2007.

[2] OMG, “Model Driven Architecture”, http://www.omg.org/mda/,2001.

[3] XMI, http://www.omg.org/technology/XML/index.htm.

[4] 太田将吾, “ソフトウェアアーキテクチャからプログラムコードへの自動変換に関する研究,” 南山大学大学院数理情報研究科 2007 年度修士論文要旨集，pp. 34-37，March 2009.