

組み込みソフトウェア向け UML における状態遷移表・ソースコード変換方式の実装と評価

小澤 陽平[†] 細川 卓誠^{††} 小泉 寿男[†]

1 はじめに

近年の組み込みソフトウェア開発では、制御だけに限らず、データ処理、音声や動画の再生、ネットワークへの対応、GUI 等により複雑化してきている。一方、大規模化しているにもかかわらず、開発期間は短期化する傾向にある。そのため、組み込みソフトウェアにも UML を用いる試みがなされている。しかし、本来 UML が用いられてきたビジネス系アプリケーションでは、データ処理が中心であることにに対し、組み込みソフトウェアでは制御が主体である。これに対し、シュレイヤー・メラー法⁽¹⁾ や実践原則をまとめた eUML⁽²⁾ が提案されている。

しかし、これらは組み込みソフトウェア開発における UML 分析のガイドラインを明確化を目的としたもので、分析・設計のギャップを埋める事の解決とはなっていない。

筆者らは、組み込みソフトウェアの UML 適用という問題に対して XUC 図 (eXtended Use Case: 拡張ユースケース図) を提案した^{(3) (4)}。XUC 図は、ユースケース図を組み込みソフトウェア向けに拡張したもので、新たにコントローラとアクションと呼ばれる要素を追加している。これにより、システムの機能要件を定義する過程を通じて分析と設計をシームレスに行うことを目指している。

本稿では、XUC 図がコントローラの機能補完のために状態遷移表を用いる事に着目する。特に XUC 分析の評価の中で、XUC 図から状態遷移表の生成について行う。また、実装面での機能追加のために、XUC 図から状態遷移表 XML を用いて XUC 図を元に記述した状態遷移表からソースコードを生成する。具体的には、XUC 図を元に記述した状態遷移表を XML で記述し、状態遷移表を記述した XML ファイルを状態遷移表/ソースコード生成機構に通す。これにより、ソースコードのスケルトンを生成する方式である。

2 XUC 図から状態遷移表生成の評価

XUC 図は、コントローラの機能補完のために主として状態遷移表を用いる。状態遷移表生成には、XUC 図のコントローラから状態の遷移を作成し、これに制御対象アクションを割り当てる事で状態遷移表を生成する。

本稿では、上記の評価のためにマウスのライントレースプログラムに適用する。(図 1 参照)

3 XUC 図からの状態遷移表 XML 記述

XML は、HTML のテーブルタグのように状態遷移表を記述することが可能である。そこで、状態遷移表を XML で記述する方式を提案する。XUC 図の記述からソースコードのスケルトン生成の流れを図 2 に示す。

A Method of Transrariion from StateChartTable to SourceCode in the Development of Embedded Software using UML .

[†]Department of Computers and Systems Engineering

^{††}Graduate School of Science and Engineering ,

Tokyo Denki University

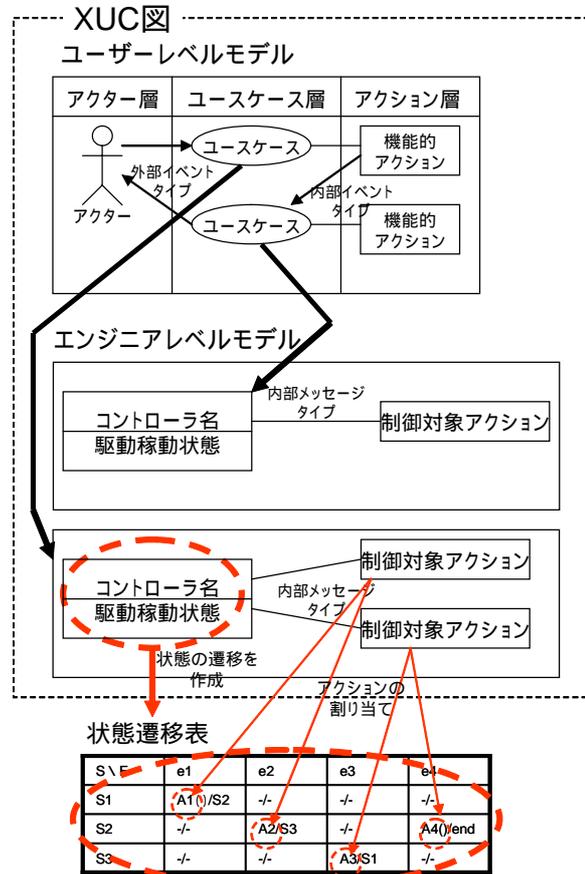


図 1 XUC 図からの状態遷移表生成の流れ

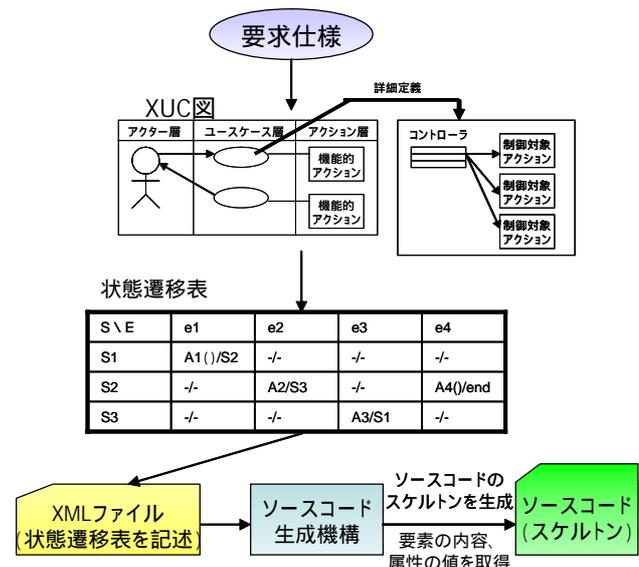


図 2 XUC 分析からソースコード生成の流れ

4 状態遷移表/XML 記述規則

XUC 図の補完のために作成した状態遷移表から，ソースコードのスケルトンを生成するために，新たに状態遷移表を XML で記述するための記述規則を提案した（図 3 参照）

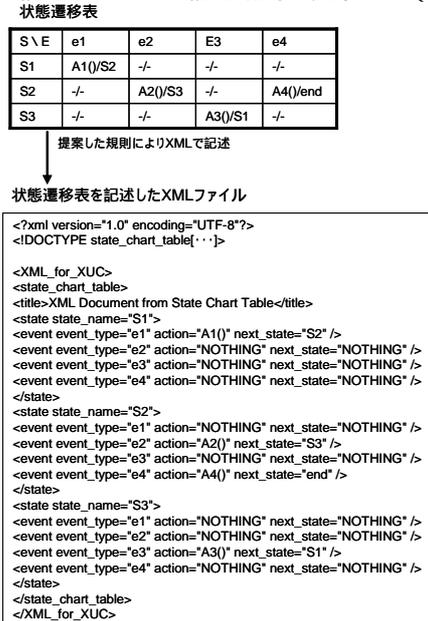


図 3 状態遷移表の XML 記述

図 3 の今回提案した状態遷移表/XML 記述規則では，HTML の TR（行）タグが XML 記述規則の state タグ，TD（列）タグが event タグに相当する．state タグの属性 name に状態名を記述し，このタグの間に event タグを記述することで，各イベントごとの動作を指定する．

5 ソースコード生成の手法

前章で記述した XML ファイルから，タグの内容，属性を取得し，ソースコードのスケルトンを生成する．図 4 に，XML ファイルからソースコード生成の流れを示す．

本手法では，XML ファイルからタグの内容，属性の値を取得するために DOM を使用する．そして，これを利用できるプログラム言語（Java）を用いてソースコード生成機構を作成する．

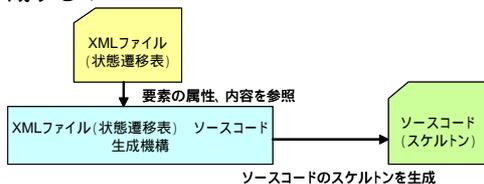


図 4 XML ファイルからのソースコード生成

6 プロトタイプ構築

上記のソースコード生成機構のプロトタイプを構築した。これは，今回新たに提案した XML/状態遷移表記述規則にもとづいて記述した状態遷移表からソースコードのスケルトンを生成するものである．

本プロトタイプは，状態遷移表の状態の遷移に着目し，これを生成する機構である。したがって，ソースコードのスケルトン単体では動作しないが，不足部分を補うことで動作可能となる．

7 評価・検証

本方式をマイクロマウスロボットであるラグウォーリア Pro のライントレースプログラムに適用する．本方式の適用により，ソースコードのスケルトンが，ライントレースプログラムの開発においてどの程度の役割を果たしたかについて評価する．

具体的には，本方式と，通常の XUC 分析から全て手書きでソースコードを記述していく方式を比較し，両者のソースコードにおいて，「スケルトン：手書き」がどの程度の割合になるかについて着目する．

また，ソースコードの「スケルトン：手書き」の割合を測定するための指標として，ステップ数を用いる．

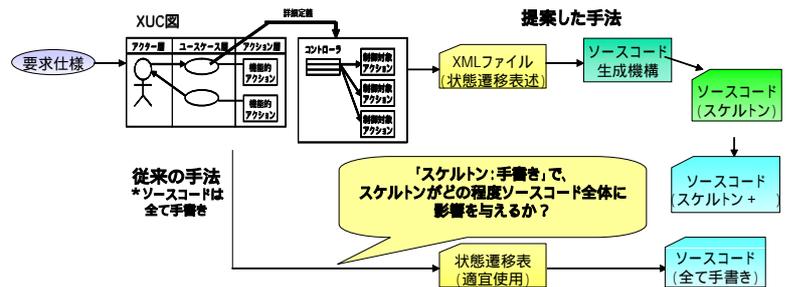


図 5 通常の方式と提案した方式の違い

6 まとめ

XUC 分析の評価をコントローラからの状態遷移表生成の点から行った．具体的には，XUC 図のコントローラから状態遷移表が記述できるかについてである．

2 つ目に，XUC 図でシステムの機能要件を記述し，これを補完するために用いる状態遷移表を XML で記述するための記述規則を提案した．3 つ目に，ソースコード生成機構のプロトタイプを作成した．また，マイクロマウスロボットのライントレースプログラムに本方式を適用し，XUC 図から状態遷移表を XML で記述した．そして，これをソースコード生成機構のプロトタイプに通し，ソースコードのスケルトンの生成を行った．

今後は，ライントレースプログラム以外の組み込みソフトウェアに本方式を適用し，ソースコードのスケルトンの果たす役割がどの程度変化するかについて評価を行う．

参考文献

- (1) Sally Shlaer, Stephen J. Mellor Object Oriented Systems Analysis(オブジェクト指向システム分析 上流 CASE のためのモデル化手法)，近代科学社，1995
- (2) 渡辺博之，渡辺政彦，堀松和人，渡守武和 記：組み込み UML eUML によるオブジェクト指向組み込みシステム開発，翔泳社，2002．
- (3) 細川卓誠，鶴見知生，岡本鉄兵，小泉寿男：組み込みソフトウェア開発におけるユースケース分析・設計方式の提案，ソフトウェア工学研究会 2003．
- (4) 小澤陽平，細川卓誠，小泉寿男：組み込みソフトウェア向け UML における状態遷移表・ソースコード生成方式，第 2 回情報科学技術フォーラム第 1 分冊．