

モバイル端末を利用したクラス図のデータ化

山田 雅人[†] 羽石 雄基[†] 高野 辰之^{††} 宮川 治^{†††}

東京電機大学情報環境学部情報環境学科[†] 関東学院大学工学部^{††}
東京電機大学情報環境学部^{†††}

1. はじめに

近年、ソフトウェア設計は再利用性を高めるために、UML を使用しクラス構造レベルでの再利用が進んでいる。具体例としてアナリシスパターンやデザインパターンの適用が挙げられる。このようなパターンをソフトウェア設計に適用するには、紙に印刷されたパターン例を参考にクラス図などからのソースコード作成を通して試行錯誤することになる。しかし、このプロセスでは手作業を避けることはできない。

そこで我々は、既存のクラス図をモバイル端末のカメラを用いて読み取り、汎用的なデータに変換するアプリケーションの開発を行った。本稿では、本アプリケーションの概要、クラス図のデータ化手順、実装を述べる。

2. アプリケーション概要

本アプリケーションは、モバイル端末のカメラで撮影した画像から既存のクラス図を読み取り、XML 形式のデータに変換するものである。クラス図とは、「クラス」というオブジェクト指向の設計単位を用いて、システムの静的な構造(モデル)を表現したものである[1]。本稿では、単一の「クラス」をクラス図として述べる。クラス図の例を図 1 に示す。クラス図は 3 つのブロックによって構成され、各ブロックにはクラス名、属性のシグニチャ(属性名, 型), 操作のシグニチャ(操作名, 返却値, 引数名, 引数の型)が記述されている。なお、クラス図が 2 つのブロックで構成される場合もあるが、本アプリケーションでは対象外とする。

3. クラス図のデータ化手順

アプリケーションがクラス図をデータ化する手順を述べる。図 2 にクラス図をデータ化する手順を示す。

まず、利用者が撮影したクラス図の画像に対

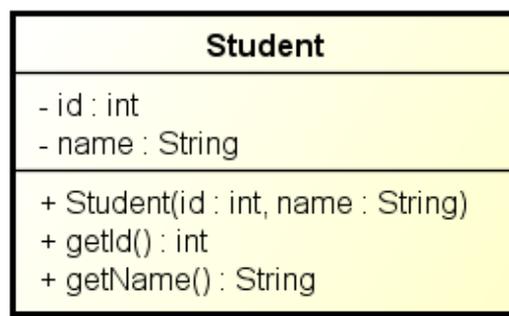


図 1 クラス図の例

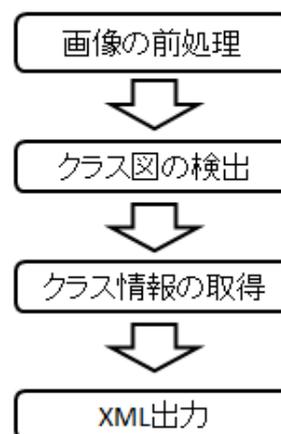


図 2 データ化の手順

して前処理を行う。次に、画像からクラス図を検出し、その後、クラス図の情報を取得する。最後に取得したデータを元に XML 化する。以下に各手順の詳細を述べる。

3.1 画像の前処理

クラス図の画像を、一定の輝度値をしきい値として 2 値化を行い、更に平滑化を行う。これにより画像からノイズを減らし、特徴検出を行いやすくする。

3.2 クラス図の検出

画像からクラス図を検出する方法は、次に示す 3 つの手順で行う。

① 図形の取得

処理された画像の中から、すべての図形を取得する。その中から十分な大きさの図形だけを選ぶ。

Digitizing Class Diagram Using Mobile Terminal
†Masato YAMADA, Yuki HANEISHI,
Graduate School of Information Environment,
School of Information Environment, Tokyo Denki University
††Tatsuyuki TAKANO,
College of Engineering, Kanto Gakuin University
†††Osamu MIYAKAWA,
School of Information Environment, Tokyo Denki University

② 矩形の検出

選ばれた図形の中から頂点が 4 つ、一つの角とその対角の角の角度が 90 度に近い図形だけを選び、その図形を矩形として検出する。

③ クラス図の検出

矩形として検出した図形に囲まれている図形を取得する。取得した図形の中から十分な大きさの図形であり、矩形として検出されたものだけを選び、選ばれた矩形の合計が 3 つであればクラス図として検出する。

3.3 クラス情報の取得

クラス情報の取得は、以下の 4 つの手順によって行う。

① 画像の回転

画像の傾きによる文字列の誤認識を軽減するため、検出したクラス図を元に、画像の回転を行う。

② 画像の分割

検出したクラス図に含まれる 3 つのブロックを、上から順に 3 枚の画像に分割する。

③ 文字の抽出

分割された画像を上から順に文字認識を行い、クラス名、属性(変数)、操作(メソッド)の情報を文字として抽出する。

④ シグニチャの取得

抽出された各文字列から、コロンやカンマなどの特定の文字をデリミッターとして分割し、型名、返却値、操作名、引数名といった情報を取得する。

3.4 XML 化

取得したシグニチャを元に XML データとして出力する。XML は多くのプログラミング言語で処理が容易な形式であり、他のシステムで読み込んで再利用することが可能である。

4. 実装

クラス図のデータ化手順に基づき実装を行った。本アプリケーションは Android OS のモバイル端末を対象としており、Android バージョン 2.3.3 以降に対応している。

前処理とクラス図の検出には、コンピュータビジョン向けライブラリである OpenCV[2]を、文字列の認識には OCR(光学文字認識)ライブラリである tesseract-android-tools[3]を使用した。しかし、OCR の認識精度は完全ではないため、誤認識が生じる可能性がある。そのため、本実装では、認識した文字列を XML 化する前に、利用者に対して認識したデータの確認画面を表示する。図 3 に確認画面を示す。利用者はこの画面を

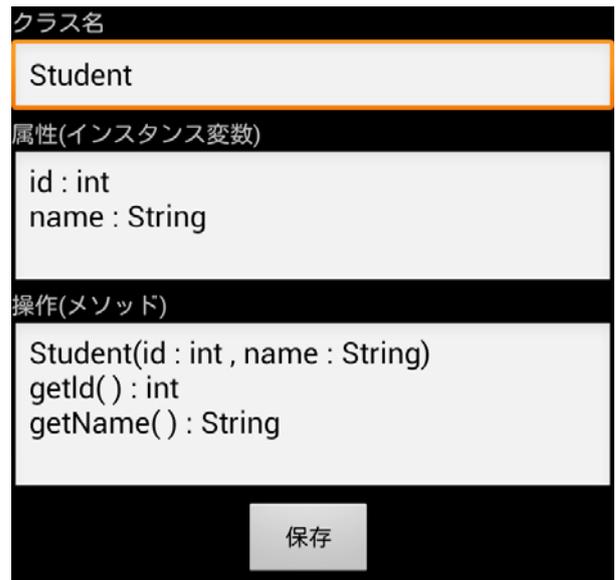


図 3 確認画面

見てクラス図が正しくデータ化されているかを確認する。確認した際にデータに誤りがあった場合、データの修正をすることができる。

そして、修正・確認が済んだら保存ボタンを押下することで、データを XML ファイルとして出力する。なお、現在の実装ではシグニチャの取得は未実装である。

5. まとめ

本研究では、既存のクラス図をモバイル端末のカメラで撮影することで、手軽にデータ化できるシステムの開発を行った。これにより、既存のクラス図の再利用を容易にさせる。

現時点では単一のクラス図の認識・データ化までを実装しているが、更に有用性を高めるために、クラス間の相関関係にも対応していく必要がある。今後は、モバイル端末のタッチパネルを活かし、保存された個々のクラスのファイルから、それぞれのクラスの間関係を直感的に編集ことができるユーザーインターフェースの実装などを目標としている。

参考文献

[1] 橋本 大輔, 小嶋 慎一, 丸山 高道ほか (著):UML 辞典, pp. 36, 翔泳社
 [2] OpenCV : 入手先<<http://opencv.org/>>(参照 2013-01-11).
 [3] Alan Viverette : tesseract-android-tools Android APIs for the Tesseract OCR library 入手先 <<http://code.google.com/p/tesseract-android-tools/>>(参照 2013-01-11).