

# プラットフォーム学習支援のためのサンプルプログラムを用いたドキュメントの関連部分抽出

Extraction of the Related Parts of Documents based on Sample Program for Platform Learning

藤浦祥雅\* 大久保弘崇† 粕谷英人‡ 山本晋一郎§

あらまし アプリケーション開発においてプラットフォームを利用する場合、プラットフォームに対する理解と知識が要求される。本稿ではプラットフォーム開発の学習者が、複数の知識源から効率よく学習を行うことのできる学習支援方法を提案する。具体的には、複数の知識源の共通基盤となる学習項目グラフを作成し、サンプルアプリケーションを用いて、それを理解するのに必要な情報を抽出する。学習項目グラフを用いることで、特定の項目に関するすべて情報を取得することができ、各知識源での情報取得を効率化する。

## 1 はじめに

近年のソフトウェア開発では生産性の向上のために、アプリケーションの土台であるプラットフォームが利用される。プラットフォームとは、アプリケーションを動作させる際に基盤となるソフトウェアであり、アプリケーションに基本的な機能を提供する。アプリケーション開発においてプラットフォームを利用する場合、プラットフォームに対する知識と理解が要求される。プラットフォームに対する知識の獲得元には、プラットフォーム提供者による公式文書、第三者による文書、プラットフォームの応用アプリケーションのソースなどがある(以降、知識源)。これらの知識源にはそれぞれ長所短所があるが、特に、異なる知識源の間の関連が示されていないことが、知識源を併用し効率的に学習する際の障害となる。

本稿では、プラットフォームを利用してソフトウェア開発を行う学習者が、複数の知識源から効率よく学習するための支援方法を提案する。本手法は二つのアイデアによって構成される。一つ目は、学習すべき項目を階層的に抽出し、その関連文章を整理することである。このことにより、分散された情報を共通項目によってまとめることができ、目的の項目に関する情報を網羅的に得ることができる。二つ目のアイデアは、サンプルアプリケーションを用いて必要な情報を抽出することである。関連文章をまとめるだけでは膨大な情報の量は変化せず、効率的に学習することは困難である。そのため、目的とする動作または機能を含むサンプルアプリケーションを用いて、学習者が扱う情報の量を減らすことを考える。サンプルアプリケーションの理解に必要な項目だけを提示することで、学習者が今必要な分の情報のみ扱うことが可能となる。本稿では、ケーススタディとして携帯電話プラットフォームである Google Android[1](以降 Android)へ適用し、本手法の評価と考察を行った。

本稿の構成を以下に示す。2章ではプラットフォームの学習と知識の獲得元について述べ、3章では本研究の手法であるドキュメントの体系化と知識源との活用方法を述べる。4章では携帯電話プラットフォームである Android への本手法の適用と本手法の考察を行い、最後にまとめと今後の課題を述べる。

\*Yoshimasa Fujiura, 愛知県立大学大学院 情報科学研究科, fujiura@yamamoto.ist.aichi-pu.ac.jp

†Hiroataka Ohkubo, 愛知県立大学 情報科学部, ohkubo@ist.aichi-pu.ac.jp

‡Hideto Kasuya, 愛知県立大学 情報科学部, kasuya@ist.aichi-pu.ac.jp

§Shinichiro Yamamoto, 愛知県立大学 情報科学部, yamamoto@ist.aichi-pu.ac.jp

## 2 プラットフォームにおける学習

プラットフォームを利用するために、開発者はプラットフォームに関して学習する必要があり、複数の知識源から情報を得る。本章では、プラットフォームを学習するために必要な情報と情報の獲得元である知識源について述べる。

### 2.1 プラットフォームを学習する上で必要な情報

プラットフォーム上で動作するアプリケーション開発方法を学習するために、以下の情報を得る必要がある。

1. アプリケーション構造
2. プラットフォームが提供する機能と実装 (API) の関係
3. 提供される機能を正しく利用するために必要な実装間関係

#### 2.1.1 アプリケーション構造

プラットフォーム上で動作するアプリケーションは、特定の構造 (deploy の構造) を持つ場合が多い。学習者はその特定の構造を理解し、構造と合致したアプリケーションを開発しなければならない。図 1 と 2 に特定の構造を持つアプリケーションの例として Android と Firefox add-on のアプリケーション構造を示す。四角で囲まれているのがディレクトリ、囲まれていないものはファイルである。Android では、Java ソースを src ディレクトリに、リソースファイルを res ディレクトリに配置する。また Firefox では、インストール時の設定を install.rdf に記述する。

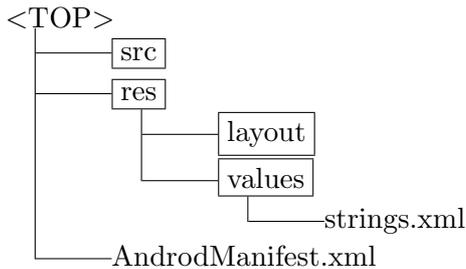


図 1 Android アプリケーションの構造の例

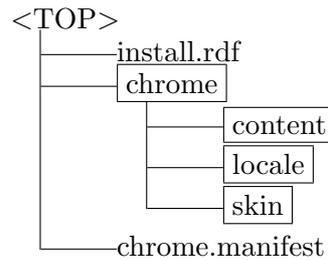


図 2 Firefox add-on の構造の例

#### 2.1.2 機能と実装の対応

プラットフォームが提供する機能を利用するとき、学習者は提供されている機能がどのように実装されているかを理解しなければならない。たとえば Android では、シングルクリックによる押下というボタン機能が Java の Button クラスとして実装されている。この機能を使うためには、学習者は Button クラスがボタン機能の実装であることを理解し、Button クラスの利用方法を学習する必要がある。

#### 2.1.3 実装間関係

プラットフォームが提供する機能とその実装の関係を理解したあと、学習者はその実装をどのように利用するか (アプリケーションのどこに加えるのか) について学習する必要がある。提供される機能をアプリケーションで利用するためには、実装間関係を理解して正しく配置しなければならない。例えば、ボタンを使うには画面機能の実装 (Activity クラス) と関係させなければならない。画面機能にボタン機能を追加するとき、学習者は Activity クラスと Button クラスの関係を理解し、Activity クラスに Button クラスを付加する方法を学習する必要がある。

## 2.2 知識源の特徴

学習者が必要とする情報は、公式文書と応用アプリケーションと第三者が作成した文書のような知識源から得る。

### 2.2.1 公式文書

プラットフォーム提供者はプラットフォームと共にプラットフォームに関する公式ドキュメント (以降ドキュメント) も提供する。プラットフォーム提供者から提供されるため、信頼できる情報といえる。

ドキュメントは網羅的ではあるが、辞典として用いる reference であり、学習順序との関連性が低く、学習に用いるには適さない場合が多い。この問題を補うために、学習順序を考慮したチュートリアルが付随する場合もあるが、チュートリアルが扱う範囲は限定されている。また、辞典的に整理されているため、機能に着目するとそれについて説明している文章が分散してしまう問題がある。以下にボタンの機能についての例を示す。図 3 は、公式文書の章立てを図示したものである。

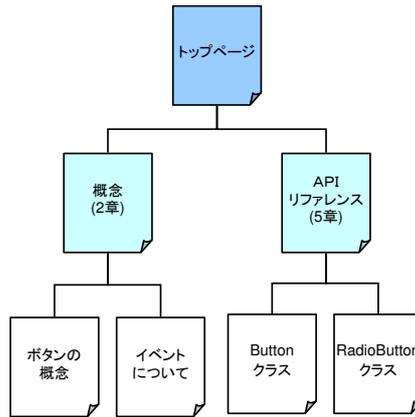


図 3 ドキュメントの構造 (ボタンの例)

概念に関する章 (2 章) と API リファレンスに関する章 (5 章) でボタンを説明している。ボタン機能に関する情報が異なる章に分散されているため、ボタン機能を学習する際、関連文章の探索が困難であり見落としも起こる。また公式ドキュメントは網羅的であり情報量が膨大であるため、ドキュメントすべてに目を通して必要とする情報を探索することは多くの時間を必要とする。

### 2.2.2 応用アプリケーション

プラットフォームを学習するために、プラットフォームの応用アプリケーションを利用する方法でも学習できる。応用アプリケーションには、プラットフォーム提供者からサンプルアプリケーションや第三者が作成したアプリケーションなどがある。

応用アプリケーションを利用した学習方法では、応用アプリケーションのソースプログラムを構造や実装 (API) の利用方法の例として扱うことで学習を行う。また応用アプリケーションを動作 (実行) させ、機能の振舞いを学習することもできる。しかし、対象とする応用アプリケーションの規模が大きくなるほど、目的の機能とソースプログラム内の対応する部分の特定が困難となる。またすべての機能に対して適切な応用アプリケーションが存在するとは限らない。そのため応用アプリケーションを利用する方法では、網羅的に学習することはできない。

### 2.2.3 第三者が作成した文書

上記以外に第三者が作成した文書で学習する方法もある。第三者が作成した文書には、技術者の blog のエントリやメーリングリスト上の QA のログ、掲示板などような Web サイトや書籍がある。第三者が作成した文書では、ある機能について説明や第三者が経験した問題などが記述される。

第三者が作成した文書の多くは、特定の機能ごとに記述されていることも多く、機能ごとにさまざまな例や情報を得ることが容易である。複数人が同じ機能について説明していることもあり、異なる観点から学習することもできる。また第三者が

陥った問題やその解決策から公式文書から得られない情報を得ることができる。しかし第三者がプラットフォームに関する情報すべてに対しての情報を提供するわけではないため、第三者が作成した文書だけで全機能の学習や機能の仕様などの詳細な情報も得ることができるとは限らない。また第三者が作成した文書であるため情報の信憑性を確認する必要がある。

### 2.3 知識源間の連携

複数の知識源にはそれぞれ長短があるため、学習者はそれらを併用して効率的に学習を行うことが望ましい。学習者の目的や方法により違いはあるが、概ね次のような順序で上記の知識源を連携させてプラットフォームの学習を行うことが効率的であると考えられる。応用アプリケーションに利用したい機能がある場合では、

1. 応用アプリケーションの振舞いから提供される機能を把握
2. 機能からソースプログラム中の対応する場所を特定
3. 必要に応じ、第三者が作成した文書を用いて機能とその実装について学習
4. その実装についての具体的な利用方法を公式文書を用いて学習

という順序であり、第三者が作成した文書内から利用したい機能を探す場合では、

1. 第三者が作成した文書を用いて目的の機能を特定
2. 第三者が作成した文書を用いてその機能の実装を把握し学習
3. 必要に応じ、応用アプリケーションのソースを用いて利用例を学習
4. その実装についての具体的な利用方法を公式文書を用いて学習

という順序でプラットフォームの学習を行う。

上記の順序のように、応用アプリケーションや第三者が作成した文書から目的の機能を特定し、その実装の利用方法などを学習する。これらの学習方法から得た情報だけでは十分でない場合、公式文書を利用して詳細な情報を得る。このように、振舞いから機能を特定したり、部分的学習が容易な第三者が作成した文書から機能を特定し、さらに詳細な情報を得る場合のみ公式文書を用いることで、学習コストを削減することができる。

### 2.4 知識源の連携における問題点

上記のように、学習者は第三者の情報を軸に学習を行う。しかし利用するプラットフォームが公開されたばかりであったり、利用者が少ない場合、第三者から提供される情報は少ない。このとき学習者が利用可能な知識源は、プラットフォームに同梱される公式文書とサンプルアプリケーションのみとなる。そのため、第三者情報を利用することで補っていたドキュメントの欠点が露呈し、学習者はドキュメントの膨大な情報を扱わなければならない。いくらかのサンプルアプリケーションも提供されているが、それらがカバーできる範囲は狭く、第三者情報の欠落した状態での学習は困難となる。

第三者情報が存在する場合においても、複数の知識源から特定の機能に関する情報を集めるのは困難となる。第三者情報もしくは応用アプリケーションから目的の機能を特定できたとしても、知識源から情報を得るときには各知識源に対して探索を行う必要がある。常に各知識源に含まれるすべての関連文章を探索できるとは限らず、見落としや探索のために多くの時間を必要とする。そのため複数の知識源の連携して学習する方法では、注目する項目に関する情報収集が問題となる。

## 3 ドキュメントの体系化と知識源の連携

本稿では、前章で述べた問題点である、新規プラットフォーム利用における第三者情報の欠落と知識源の連携効率化に対処するために、学習項目グラフを用いた学習方法を提案する。

### 3.1 概要

ドキュメントとソースプログラムの連携に関する研究 [12–14] は多く存在するが、本研究では、与えられたサンプルアプリケーションを理解するための知識を過不足なく提示するために、サンプルアプリケーションからそれを理解するためのドキュメントを自動的に抽出することを目的としている。そのため本研究では、学習項目グラフによる関連情報の整理とサンプルアプリケーションを用いた必要情報の抽出という二つのアイデアを提案する。

学習項目グラフによる関連情報の整理では、プラットフォームが提供する機能を学習するのに必要な項目を学習項目として特定し、各項目をアプリケーションの構造を軸として構造化を行う。各学習項目を公式ドキュメントの関連文章と関係付けることで公式ドキュメントを学習に適した形式で体系化する。本研究では体系化したグラフを「学習項目グラフ」と呼ぶ。この学習項目グラフを作成することで、分散された情報を学習項目により整理する。

作成した学習項目グラフのままでは膨大な情報量は変わらないため、サンプルアプリケーションを用いて学習者が扱う情報を減少させることを考える。学習に利用したい応用アプリケーションを解析し、解析した情報を用いて学習項目グラフから対象とする応用アプリケーションを学習するのに必要な情報のみを抽出する。このことにより、学習者は今必要な分の情報だけに注目することが可能となる。

学習項目グラフは予め人手により 1 回だけ作成する。その後、与えられたサンプルアプリケーションを解析し、学習項目グラフを自動で tailoring する。

### 3.2 学習項目グラフによる関連情報の整理

#### 3.2.1 構成要素

プラットフォームが提供する機能はさまざまな形式で実装され、本研究ではその実装を「構成要素」と呼ぶ。実装方法は API であったり、プラットフォームで指定された特定の名前を持つディレクトリやファイルなどである。それぞれの例には Button クラス、Android の res/values ディレクトリ (図 1)、Firefox add-on の Chrome.manifest (図 2) が挙げられる。これらをアプリケーションのソース内で利用することにより、提供される機能を利用することができる。

#### 3.2.2 学習項目

構成要素を学習する際、構成要素に関する複数の項目を学習する必要がある。本研究ではその項目を「学習項目」とし、機能を複数の学習項目に分割する。すべての学習項目を学習することで、学習者は機能を完全に理解できるものとする。ボタン機能を用いて例を示す。ボタンを学習するには、以下のような学習項目がある。

- ボタンについての概念
- ボタンが押されたときのイベント
- ボタンの実装である Button クラスの利用方法

学習者はボタン機能を使う際、ボタンの概念やボタンが押されたときのイベントなどの概念に関する内容と、Button クラスのメソッドなどの実際の利用方法に関する内容を学習しなければならない。また学習項目は、図 4 のように構成要素と複数の学習項目で階層的に構成される。

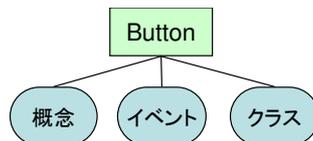


図 4 構成要素と学習項目の関係

### 3.2.3 ドキュメントの体系化

上記で定義した構成要素と学習項目を用いてドキュメントを再構成し、プラットフォームに関する情報の体系化を行う。以下に学習項目グラフの作成手順を示す。また図5に学習項目グラフの例である。

1. 構成要素をアプリケーション構造で配置する
2. 各構成要素に対し、学習項目を配置する
3. 各学習項目に対し、公式文書の関連文章を対応付ける
4. 構成要素間の関係を付加する

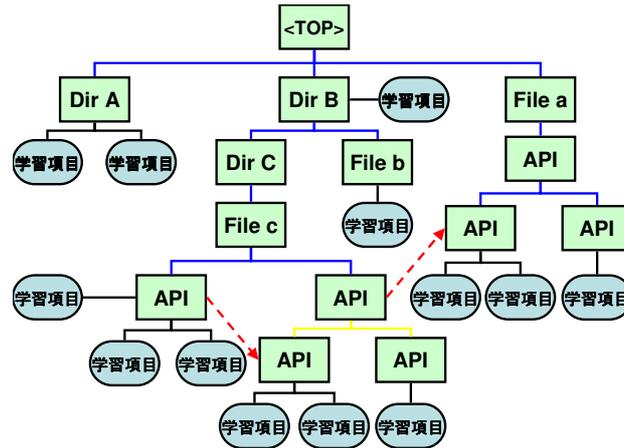


図5 学習項目グラフの例

学習項目グラフのノードは構成要素と学習項目ある。学習項目グラフのエッジは、is-a 関係 (黄)、member-of 関係 (青)、呼び出し関係 (赤) の三種類存在する。学習項目グラフは、オントロジー [2, 3] と類似しており、is-a 関係と member-of 関係はオントロジーで定義された関係を利用した。

is-a 関係は、2つの構成要素間の関係が継承関係を持つ場合の関係である。

member-of 関係は、2つの構成要素間の関係が一方を構成する要素である場合の関係である。アプリケーションは、XML など構造を持った言語が利用される場合やディレクトリ構造を持つ場合がある。member-of 関係では、構造を用いて構成要素間の関係を表現する。

呼び出し関係は、2つの構成要素間の関係が一方を呼び出す要素である場合の関係である。文献 [4] では、ライブラリ関数に関するノウハウをライブラリ関数間の呼び出し関係から取得できることを示している。この考え方は、ライブラリ関数だけでなく、本研究の構成要素間の呼び出し関係でも成り立つ。アプリケーションはある構成要素が別の構成要素を呼び出すことにより、機能を利用している。

ある構成要素を呼び出し可能な構成要素はある程度特定することができるが、呼び出し関係は無数に存在するため、対象とするアプリケーションを解析することで、対象のアプリケーションごとで呼び出し関係の取得と表示を行い、学習者に対しノウハウを提供する。また呼び出し関係は、他の2つの関係とは異なる性質も持つため、学習項目グラフでは、エッジの種類を変えて表示する。

### 3.3 サンプルアプリケーションを用いた必要情報の抽出

本研究では前節までで定義した学習項目グラフに、対象とする応用アプリケーションの解析データを加えることで、そのアプリケーションの学習に必要な分だけの情報を抽出する。

対象のアプリケーションからは、利用している構成要素と構成要素間の呼び出し関係を取得する。学習項目グラフから利用している構成要素のみを抽出し、構成要素間の呼び出し関係を付加することで、その関係を表現する。このことにより公式ドキュメントの膨大な情報を今対象とするアプリケーションの学習に必要な分だけに減少させることができる。また呼び出し関係を付加することで、各構成要素間の関係の理解を支援する。

#### 4 Google Android への適用

本稿ではケーススタディとして携帯電話向けプラットフォームである Android への適用を行った。Android は Google が 2007 年 11 月に発表したオープンなソフトウェアプラットフォームであり、本研究が想定する第三者からの情報が不足しているプラットフォームとして適切であると考えられる。

Android アプリケーションは、Java 言語と XML で開発を行う。Android が提供する構成要素の数を表 1 に示す。Android プラットフォームの公式文書は HTML 形式で提供されており、すべて英語で記述されている。表 2 に各章の内容と規模を示す。表 1 と表 2 は、2008 年 11 月に公開された Android SDK 1.0 r2 に関する表である。本稿では、Android SDK 1.0 r2 に関して適用を行った。

表 1 Android の構成要素数

Java クラス数	595
パッケージ数	40
ディレクトリ数	9
ファイル数	7

表 2 Android のドキュメントの内容と規模

章番号	内容	サイズ (word)	ファイル数
1	Android プラットフォームの構成	843	1
2	インストール, 開発方法, チュートリアル	24,178	14
3	UI や構成ブロックなどの説明	244,790	21
4	Android アプリケーションの設計理念など	14,813	13
5	API のリファレンス	1,906,335	2517
6	Android に関するよくある質問とその回答	10,494	7

また適用を行うために支援ツールを開発した。図 6 に支援ツールの外観と図 7 の構成を示す。

本支援ツールの外観は三つの部分に分かれている。左上の取得情報では、対象とするアプリケーションの構成や利用されている構成要素の一覧、各構成要素の学習項目一覧が表示される。また学習項目を選択すると Web ブラウザでドキュメントの関連ページを表示する。

左下の縮小グラフでは、右側に表示される学習項目グラフの縮小版が表示される。この縮小グラフを用いて学習項目グラフ全体の把握や構成要素の探索を助ける。

右側には学習項目グラフが表示される。本支援ツールではプラットフォーム全体の学習項目グラフ、対象アプリケーションの学習項目グラフ、マーキングされた学習項目グラフの 3 種類を提供する。プラットフォーム全体の学習項目グラフはすべての学習項目を含む学習項目グラフであり、対象アプリケーションの学習項目グラフは対象のアプリケーションに含まれる構成要素のみで作成した学習項目グラフで

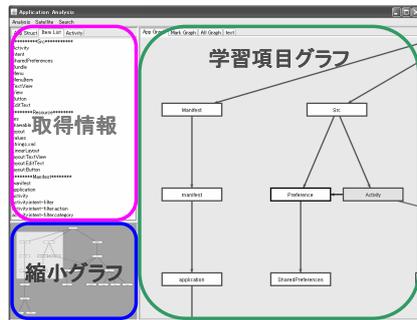


図 6 支援ツールの外観

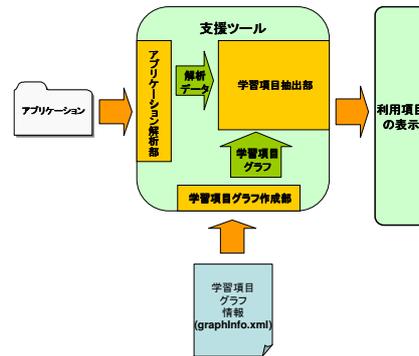


図 7 支援ツールの構成

ある．マーキングされた学習項目グラフはプラットフォーム全体の学習項目グラフの構成要素の内，対象アプリケーションに含まれる構成要素をマーキングした学習項目グラフである．

また支援ツールはアプリケーションと学習項目グラフ情報 (4.1 節) を入力とし，上記の情報を出力する．支援ツールはアプリケーション解析部，学習項目グラフ作成部，学習項目グラフ抽出部の三つの部分で構成されている．アプリケーション解析部では，対象アプリケーションで利用されている構成要素などを取得する (4.2 節)．学習項目グラフ作成部では，学習項目グラフ情報からプラットフォーム全体の学習項目グラフを作成する．二つの部分から得られた解析データと学習項目グラフを用い，学習項目抽出部で対象アプリケーションの学習項目グラフを出力する．

#### 4.1 Android での学習項目

本ツールは，主要な構成要素 94 種類の学習項目のみを対象とする．この 94 種類は，公式に提供される 8 つサンプルアプリケーションと 4 つの第三者が作成した文書 (Web サイト) で利用される構成要素のうち 3 つ以上で利用されている構成要素を頻繁に利用される構成要素として選定した．表 3 と表 4 に調査対象の詳細を示す．また表 5 に構成要素の分類を示す．

表 3 調査対象 (サンプルアプリケーション)

サンプルアプリケーション名	内容
HelloActivity	Activity の HelloWorld
NotePad	メモ帳アプリケーション，データベースを利用
SkeletonApp	アプリケーションのスケルトン，画像やイベントなどを利用
Snake	ゲームアプリケーション
LunarLander	ゲームアプリケーション
HOME	待ち受け画面のサンプルアプリケーション
SoftKeyborad	キーボードアプリケーション
NotePad Turtorial	メモ帳アプリケーション，データベースを利用

本支援ツールでは，学習項目グラフ情報を XML ファイル (graphInfo.xml) で提供する．graphInfo.xml は手作業で作成する．

#### 4.2 フィルタリング

本支援ツールでは，対象のアプリケーションから情報取得とフィルタリングを自動で行う．アプリケーションは，次の方法で解析を行った．

表 4 調査対象 (Web サイト)

Web サイト名	内容
JavaDrive [5]	さまざまな View やレイアウト中心のサイト
Android メモ [6]	UI からデバイス制御まで広範囲のソースを公開しているサイト
Android 実習マニュアル [7]	基本的な構成要素の詳細を記述
山本研 Tutorial [8]	藤浦が作成した二つのチュートリアル

表 5 構成要素の分類

分類	個数
API (クラス)	39
API (XML)	38
ディレクトリ	9
ファイル	7

Java ソース Import 宣言から利用しているクラスを抽出

XML ファイル 利用している XML 要素を取得

ディレクトリとファイル 利用しているディレクトリとファイルを抽出

上記の方法で取得した構成要素と静的解析により取得した呼び出し関係を用い、アプリケーションごとの学習項目グラフを出力する。

#### 4.3 本手法の評価と考察

本節では Android へ適用した本支援ツールから本手法の評価と考察を行う。本支援ツールでは、サンプルアプリケーションと学習項目グラフ情報を入力とし、アプリケーション構造を軸とした学習項目グラフを出力する。

JavaDoc [9], Doxygen [10], perldoc [11] などの API ドキュメントのリファレンスツールでは、アプリケーション構造を学習することは困難であり、公式ドキュメントから学習する必要がある。本手法では学習項目グラフをアプリケーション構造を軸に体系化するため、未知の構造を知ることができる。たとえば Android アプリケーションでは、図 1 の構造の他に `res/value/arrays.xml` という配列の情報を記述するファイルがある。このファイルは API リファレンスには含まれないため、学習者は配列を管理するファイルの存在をドキュメントを読むことや `arrays.xml` を利用したソースと出会うまで知ることはできない。本手法では、学習項目グラフを用いることで学習者に `arrays.xml` の存在を明示する作用がある。

また各構成要素において学習項目の一覧を表示するため、学習者が知らない情報も把握することができる。各学習項目はドキュメント内の関連ページと対応しているおり、本支援ツールから参照することが可能である。そのため学習者は構成要素の関連ページを探索する必要がなく自分が必要とする学習項目を選択することができる。構成要素で学習項目をまとめているため、必要なページの見落としを防ぐことができる。

さらに本支援ツールでは、選択された構成要素が利用されているファイルやディレクトリを表示する。そのためアプリケーション内から構成要素の位置を特定することを助け、学習項目グラフとサンプルアプリケーションを結びつける。そして学習項目グラフからドキュメントを参照することで、サンプルアプリケーションの学習を効率的に行うことができる。

上記のように本手法の学習項目グラフを用いることで、サンプルアプリケーションとドキュメントの欠点である、ソース内の位置特定とドキュメント内の探索を助け、二つの知識源を有効に利用することが可能となったと考える。また既存のツールでは困難なアプリケーション構造や概要を学習に対しての支援も可能であり、多

言語 (Android では Java と XML) の対応する．このことから本手法は複数の知識源を連携させ、プラットフォームを効率的に学習することができるといえる．

## 5 おわりに

本稿では、プラットフォーム開発の学習者が、複数の知識源から効率よく学習を行える支援方法を提案した．具体的には、学習項目を用いて複数の知識源に含まれる関連情報をまとめ、アプリケーション構造を軸に学習項目グラフとして体系化した．このことにより分散していた情報を整理し、特定の機能に関する情報の探索を効率化した．また応用アプリケーションの理解に必要な分のドキュメントを抽出することで、学習者が扱う情報を減少することが可能となった．本手法を用いることで、公式文書と応用アプリケーションという二つの知識源を連携させ、効率的に学習を行うことが可能となる．

本稿では、公式文書と応用アプリケーションを基に作成してきた．これはプラットフォームに関する情報が一番少ない状況を想定しているためである．このとき利用できる知識源は、プラットフォーム提供者が提供する公式ドキュメントとサンプルアプリケーションのみであり、第三者からの情報が少ない．しかし多くのプラットフォームでは第三者からの情報が豊富であり、有益な情報が多く存在する．本研究では第三者が作成した文書や応用アプリケーションとも連携することができる．方法としては、学習項目と第三者が作成した文書内の関連文書と対応付けるだけである．こうすることで、ある学習項目から公式文書と第三者が作成した文書の両方を参照することが可能となる．また応用アプリケーションでは、ソースが提供されているのであれば、利用可能である．

現在学習項目グラフの情報は手動で作成しているが、その情報はすべて公式文書内に存在するため自動もしくは半自動的に作成できることが望まれる．またアプリケーション解析方法や提示する学習項目の選択など、学習者が必要としない情報をより削減する方法の考察が今後の課題である．本支援ツールでは公式文書を HTML 形式とし、学習項目からの参照をページ単位で行っている．しかしページ内には複数の学習項目が存在し、目的の学習項目をより明確に示すため、ページ内の必要部分を抜き出す必要がある．

## 参考文献

- [1] Google Android: <http://code.google.com/android/>
- [2] 溝口理一郎, 池田満: オントロジー工学序説 - 内容指向研究の基盤技術と理論の確立を目指して, 人工知能学会誌, Vol. 12, No. 4, pp. 559-569, 1997.
- [3] 溝口理一郎, 池田満, 来村徳信: オントロジー工学基礎論 - 意味リンク, クラス, 関係, ロールのオントロジー的意味論 -, 人工知能学会誌, Vol.14, pp.1019-1032, 1999.
- [4] 渥美 紀寿, 山本晋一郎, 阿草清滋: 関数呼び出し依存グラフに基づくプログラミング支援, 日本ソフトウェア科学会, 2001
- [5] JavaDrive: <http://www.javadrive.jp/android/>
- [6] Android メモ: <http://www.saturn.dti.ne.jp/~npaka/android/>
- [7] Android 実習マニュアル: <http://tutorial.jp/prog/android/andman.pdf>
- [8] 山本研 Tutorial: <http://www.aichi-pu.ac.jp/ist/lab/yamamoto/android/android-tutorial>
- [9] JavaDoc: <http://java.sun.com/j2se/javadoc/>
- [10] Doxygen: <http://www.doxygen.org/>
- [11] perldoc: <http://www.perldoc.com/>
- [12] 後藤英斗, 大久保弘崇, 粕谷英人, 山本晋一郎: 文脈に基づいたソースプログラムとドキュメント間の識別子の対応付け手法, 情報処理学会研究報告. ソフトウェア工学研究会報告, Vol.2005, No.29(20050317) pp. 41-48, 2005
- [13] 古山将佳寿, 山本晋一郎, 阿草清滋: ドキュメントを含むソフトウェアモデルの提案, 日本ソフトウェア科学会, 1999
- [14] 大場勝, 権藤克彦: アスペクト指向を用いたドキュメント整理法の提案, 日本ソフトウェア科学会第7回プログラミングおよび応用のシステムに関するワークショップ, 2004