Vol.2013-CSEC-62 No.63
Vol.2013-SPT-6 No.63
2013/7/19

Android アプリの静的解析を用いた利用 API と外部モジュール検知によるプライバシーポリシー作成支援機構

磯原隆将[†1]　川端秀明[†1]　竹森敬祐[†1]　窪田　歩[†1]

スマホアプリによる利用者情報の取得と送信が，プライバシー上の懸念となっている．これを解決する手段として，アプリによる利用者情報の取り扱いを的確に説明する，アプリのプライバシーポリシー（以下，アプリプラポリ）を用いて，利用者の承諾を得ることが推奨されている．我々はこれまで，XML 文書を用いて，総務省のスマートフォン プライバシー イニシアティブが推奨する 8 項目を備えたアプリプラポリを管理・運用する手法を提案している．しかし，これは，アプリプラポリの形式的な要件を満たす提案であるものの，記述の正確性といった内容的な要件を満たすものではなかった．そこで本研究では，記述内容が正確なアプリプラポリの作成を支援する機構を提案する．具体的には，アプリのプログラムを静的解析して，利用している API と外部モジュールを検知することで，アプリプラポリへの記載が必要な情報を提示する．提案のツールは，アプリの開発工程を考慮して，開発環境向けプラグインとモバイルアプリのツールを設計する．Android アプリ向けのプロトタイプを試作し，提案のツールがアプリプラポリの内容的な品質の向上に寄与することを示す．

Privacy Policy Generation Assistant-mechanism based API and External Module detection for Android Application

TAKAMASA ISOHARA[†1]　HIDEAKI KAWABATA[†1]
KEISUKE TAKEMORI[†1]　AYUMU KUBOTA[†1]

Collecting and sending of privacy related information via mobile applications is a big privacy issue on mobile security. To dispel this privacy concern, making an agreement between an application provider and an application consumer by using an application privacy policy is a practical solution. Thus, we have proposed XML-based mobile application privacy policy management scheme. Although our proposal has contributed to improve a formal quality of an application privacy policy, description content has not been improved yet. In this paper, we propose a privacy policy generation assistant mechanism. Proposed mechanism uses a static inspection technique to find out implemented APIs and external modules in a target program. Inspection result provides useful information to write a correct application privacy policy. We implement prototype program in form of an Eclipse plug-in and an Android application. We show that our proposal is useful assist to generate a correct application privacy policy.

1. はじめに

スマホアプリによる勝手な情報送信がプライバシーの観点から問題視される中で，2012 年 8 月に総務省から，スマホ向けアプリの透明性の向上や，必要最低限の情報送信にすべきこと，アプリ事業者や Market 運営者の果すべき役割など，利用者のプライバシー保護の目指すべき姿として，「スマートフォン プライバシー イニシアティブ（以後，SPI）」がリリースされた[1]．特に，送信される情報・その目的・送信先などに関して，解りやすく説明するアプリのプライバシーポリシー（以後，アプリプラポリ）を策定し，利用者から承諾を得ることが推奨されている．また，2012 年 10 月には，モバイルコンテンツフォーラム（MCF）から，SPI を参考にしたアプリプラポリの執筆例が公開された[2]. EU では忘れられる権利[3], 米国では Do Not Track[4] が提唱され，ネット上でのプライバシー保護の取り組みと利用者関与の機会が推進されている. こうした流れを受け，国内の関係者が連絡協議会を通じて情報共有を行う場も設けられている[5].

我々は，アプリから実際に送信される情報の解析とアプリプラポリ中に送信される情報に関する記載の有無について，3 回／3 年間の調査を実施してきた．直近の調査では，63%のアプリが端末や SIM に付随する固定の ID（以後，グローバル ID）もしくはプライバシー情報を送信することが解り，うち 11%のアプリしかアプリプラポリで送信情報を正しく説明していないことが解った．送信される情報の多くは，アプリ本来の趣旨に沿わない，過剰な情報送信が目立ち，これらについてアプリプラポリに正しく記載されないことが原因である．また，SPI が推奨する 8 項目の記載状況や，他社が調査した結果と照合を行い，多くのアプリプラポリにおいて，SPI 8 項目を満たさないこと，用語や文章が曖昧で読み解き者によって判断がばらつく問題が明らかになった．そもそも，アプリプラポリではなく，個人情報保護法に基づく事業者としてのプラポリ（以後，事業者プラポリ）を流用しているものも多く，アプリから送信される情報を適切に説明できていない[6].

本研究では，アプリが取得して外部に送信する情報に関する記述に関して，アプリの挙動と記述内容の整合性が取れた正確性と，解釈の齟齬が生じにくい明確性を実現することが，アプリプラポリの有効性を高めてゆく上での要件

†1 (株)KDDI 研究所
KDDI R&D Laboratories

Vol.2013-CSEC-62 No.63
Vol.2013-SPT-6 No.63
2013/7/19

と捉える.

上記の要件を解決する手段として，アプリプラポリを作成する際に記載すべき内容を提示する支援機構を提案する.提案の機構は，対象となるアプリのソースコードやファイルを静的解析して，アプリが利用する API と外部モジュール検知を行う．API と外部モジュールの検知結果にもとづいて，アプリが取得して外部に送信する可能性のある情報を提示することで，正確性と明確性を備えたプラポリの作成を支援する. 支援機構は, アプリの開発工程を考慮して，上流工程で利用することを想定した開発環境向けプラグインと，下流工程で利用することを想定したモバイルアプリを設計する．Android アプリケーションを対象としてプロトタイプを実装し，ツールがプラポリ作成に有益な情報を提示すること，ツールの利用を通じて，SPI の 8 項目を満たし，アプリが取得・送信する情報に関する正確性と明確性を備えたアプリプラポリの作成が促進される効果を期待できることを示す.

2. 実態調査

2.1 利用者情報と SPI8 項目

表 1 に，本研究で注目する利用者情報を纏めておく[1]. スマホ特有のグローバル ID とプライバシー情報であり，利用者に直接結び付かず，利用者側で取替えが可能な cookie やアプリ独自の ID を含めていない.

表 1　スマホにおける利用者情報

グローバル ID	International Mobile Equipment Identity (IMEI), International Mobile Subscriber Identity (IMSI), SIM serial number (ICCID), OS 生成 ID 電話番号, MAC アドレス
プライバシー情報	メールアドレス，電話帳，位置，インストールアプリ名，アプリ実行ログ（含：実行アプリ名，通話履歴など）

2.2 3 回（3 年）の実態調査

我々は，3 回／3 年に渡り，スマホ向け無料アプリから送信される利用者情報に関する実態調査を行ってきた．アプリは，日本のドメインから参照した際の人気順にダウンロードしている．結果を表 2 に示す．プラポリとの整合性に関する調査結果も示しているが，対象にしたプラポリとして，アプリプラポリと事業者プラポリの両者を読み解いた結果であることに注意されたい.

SPI は 2012 年 8 月に公開された. 表 2 の 2011, 2012, 2013 年の 3 列は，それぞれ空白期，SPI 公開前，SPI 公開後の状態であることを勘案して参照すると解りやすい.

表 2　スマホアプリからの利用者情報の送信状況

	2011.8–2012.1 調査(400件)	2012.4–2012.5 調査(100件)	2013.2–2013.3 調査(100件)
情報収集ﾓｼﾞｭｰﾙを組み込んだｱﾌﾟﾘ	56.9%*1 (558件)	82% (82件)	63% (63件)
利用者情報を送信するｱﾌﾟﾘ	45.3%*2 (181件)	81% (81件)	63% (63件)
ﾌﾟﾗﾎﾟﾘに送信情報が正しく記載のｱﾌﾟﾘ(白)	17/181件=9% 17/400件=4%	2/81件=3% 2/100件=2%	7/63件=11% 7/100件=7%
ﾌﾟﾗﾎﾟﾘに送信情報が誤って記載のｱﾌﾟﾘ(灰)	7/181件=4% 7/400件=2%	13/81件=16% 13/100件=13%	29/63件=46% 29/100件=29%
ﾌﾟﾗﾎﾟﾘが無く利用者情報を送信するｱﾌﾟﾘ(黒)	157/181件=87% 157/400件=39%	66/81件=81% 66/100件=66%	27/63件=43% 27/100件=27%

*1 この調査のみ980件のｱﾌﾟﾘを対象としている。
*2 2011.8–2012.1の調査ではSSL通信で送信される利用者情報を対象外。

2012 年 4-5 月の情報収集モジュール[7]-[13]組み込み率と利用者情報の送信率が 8 割を超えているのに対して，2013 年 2-3 月では，6 割程度になっており，SPI のリリースによって，素行が良く解らない外部の情報収集モジュールの組み込みや，不必要な利用者情報の収集を控えるアプリ事業者が現れてきたことが伺える．また，アプリもしくは事業者プラポリを持たずに勝手な情報送信をするアプリ(黒)が，87%→81%→43% へと減少している．送信情報が誤って記載されているものの，プラポリを通じて利用者に説明しようと努力が伺えるアプリ(灰)が，4%→16%→43%へと増加している．特に，SPI 公開前後で減少と増加の変化量が著しい．これより，SPI[1]や MCF ガイド[2]を参考にプラポリを書く流れが起きていることが伺える.

ここで，正しく送信情報を記載できているアプリは，9%→3%→11%と，2013 年現在も 1 割程に留まっており，プラポリを正しく書けない現状がある．主な理由は，素行の解らない外部の情報収集モジュールを組み込むことに起因している.

2.3 2013 年 4-5 月（100 件）の詳解

プラポリとして，サービスを含めたアプリプラポリと，Web サイトを含めた事業者プラポリのいずれが，アプリの説明に適用されているかの調査を実施した．結果を表 3 に示す.

アプリプラポリを持つアプリは，利用者情報の送信の有無に関わらず 25%程であることが解る．また，事業者プラポリで代用したアプリが 30%程あることも解る. ちなみに，アプリプラポリ 16 件のうち, 送信情報を正しく記載していたものは 5/16 件=31%であり, 事業者プラポリ 20 件のうち，送信情報を正しく記載していたのは 2/20 件=10%であった．これより，事業者プラポリではアプリから送信される利用者情報を適切に説明しきれていないことが解る．また，たとえアプリプラポリを持っていたとしても，送信情報を正しく説明できているアプリが少ないこともわかる.

表 3 プラポリに関する詳解（2013 年 4-5 月）

	全ｱﾌﾟﾘ(100件)	利用者情報を送信するｱﾌﾟﾘ(63件)
アプリプラポリを持つアプリ	24/100件=24%	16/63件=25% (正確5/16件=31%)
事業者プラポリを持つアプリ	29/100件=29%	20/63件=32% (正確2/20件=10%)
プラポリを持たないアプリ	47/100件=47%	27/63件=43%

次に，表 3 に示したアプリプラポリ 24 件と事業者プラポリ 29 件の計 53 件について，SPI 8 項目の記載状況を表 4 に纏める．尚，本研究では第 6 項目を第三者提供と情報収集モジュールに分けて調査を進めた．

結果として，①のアプリ事業者に関する説明は，100%のプラポリに記載があった．しかし，⑥-2 の情報収集モジュールの有無については，23%のプラポリにしか記載がない．アプリに組み込まれた情報収集モジュールについて，アプリ事業者から利用者へ説明責任があるにも関わらず，その意識が薄い様子が伺える．これに起因して，②の送信情報に関する説明が 92%のプラポリに記載がるにも関わらず，前記の通り約 9 割のアプリが，送信情報についての記載漏れが生じている．

ちなみに，SPI 8 項目の全てを記載しているアプリは 4/100 件=4%あり，うち送信情報を正しく記載できていたものは 0/4 件=0%であった．形式論も必要であるが，そもそも送信情報を正しく説明するために，アプリの挙動を技術検証し，アプリプラポリの記載内容が正しいことを第三者が客観的に検証するスキームが望まれる．

表 4 SPI の 8 項目と 53 件のアプリへの記載状況

①	情報を取得するアプリ提供者等の氏名又は名称	53/53 件=100%
②	取得される情報の項目	49/53 件=92%
③	取得方法の記載の有無（自動送信，手入力，Web 側での閲覧履歴）	39/53 件=74%
④	利用目的の特定・明示	52/52 件=98%
⑤	通知・公表又は同意取得の方法，利用者関与の方法（送信の停止，情報削除）	32/53 件=60%
⑥	1)外部送信・第三者提供の有無 2)情報収集モジュールの有無	47/53 件=89% 12/53 件=23%
⑦	問い合わせ窓口の情報	36/53 件=68%
⑧	プライバシーポリシー変更の手続きの説明	32/53 件=60%

2.4 他社の調査結果との照合

ほぼ同じ時期に，日本総研(株)によって SPI 8 項目に注目した 40 件のプラポリの読み解きが行われた．我々が対象にしていた 100 件のうち，33 件が同じアプリを調査していた．そこで，両社の結果を持ち寄り，SPI 8 項目への適合性について確認したところ，17/32 件＝52% のアプリのプラポリにおいて，いずれかの項目で読み解きの判断が分かれた．この様子を図 1 の左側（黄色）に示しておく．この原因は，プラポリの説明文や用語が曖昧なために，判断が分かれたのが主原因である．ここでは，敢えて読み解きミスとは言わないでおく．尚，図 1 の右側には，プラポリに記載のあった送信情報について我々が列挙した状況を示しておく．この送信情報の記載内容と，アプリから実際に送信される情報を検証することが望まれる．

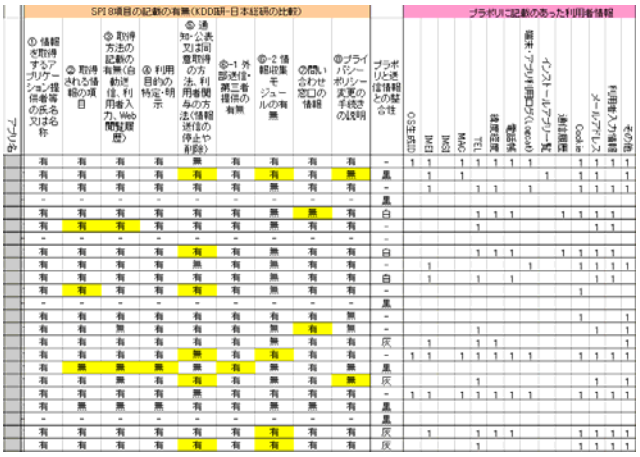

図 1 SPI の準拠性調査の様子と判断が分かれた状態
(黄色い箇所が判断の分かれた箇所)

2.5 情報収集モジュールから送信される利用者情報

図 2 に，2013 年 4-5 月調査の 100 件のアプリを 5 分間実行した際に，情報収集モジュールから送信された利用者情報の様子を示しておく．複数のグローバル ID を送信する情報収集モジュールも多い．中には，グローバル ID と位置情報を合わせて送信するものもあり，利用者の位置を活かした広告コンテンツの提供の様子が伺える．尚，Web 向け広告で利用されているような cookie を使った利用者追跡を行っているものは見当たらなかった．スマホアプリ向け情報収集モジュールで実現できる cookie 方式が 2012 年 12 月に提案されている中で[14]，広告のために忘れられる権利が侵されかねないグローバル ID を用いる理由は成り立たない．利用者のプライバシー保護の観点から，利用者に直接結びつかず，利用者によって消去や取替えが可能な cookie 方式の普及が望まれる．

3. アプリ開発の工程に着目したプラポリ作成に関する課題整理

我々はこれまで，SPI の 8 項目を満たすアプリプラポリの作成を支援する手段として，8 項目を備えたプラポリ文書記述用 XML 文書を定義し，定義した文書の作成を支援

図 2　情報収集モジュールから送信された利用者情報

するアプリを提案している [15]．この提案は，SPI の推奨する 8 項目を備えたアプリプラポリの作成を支援するものである．しかし，各項目の正確性を向上するものではなく，アプリプラポリの作成者は，[1]や[2]を参照しながら，自らの判断で各項目に適切と考えられる内容を記述する必要があった．

正確なアプリプラポリの作成が困難である理由として，以下の 2 つがある．1 つ目は，アプリ開発者が，アプリに取り込む外部モジュールの実装を把握することができないことが挙げられる．この課題について， [15]では，アプリを構成する各モジュールの提供者が，自身の開発するモジュールに関するプラポリを作成することを解決策として提案した．しかし，現状では多くの外部モジュール開発者は，提供するモジュールに関するプラポリを作成・提供していない．したがって，外部モジュールを利用するアプリ開発者が，アプリ全体について，適切なプラポリを作成可能な仕組みが必要となる．2 つ目の理由として，アプリの開発から公開までの工程において，開発の上流工程に位置してプログラムの作成に携わる担当者と，開発の下流工程に位置してアプリの公開に携わる担当者が分かれている事例が多いことがあげられる．ここで，一般的には，プログラムの実装や挙動を正確に把握している上流工程の担当者は，プログラムの作成のみに携わることが多く，プラポリの作成は，下流工程の担当者が実施する事が多い．このとき，下流工程の担当者はプログラムの実装や挙動を正確に把握できないまま，プラポリ作成を行わざるを得ない．

4. プラポリ作成支援ツールの設計

アプリ本体と外部モジュールを含むアプリ全体に関する適切なプラポリ作成を支援する手段として，アプリの静的解析によって，正確性と明確性を備えたアプリプラポリの作成に有効な支援情報を提供するツールを設計する．

4.1 プラポリ作成支援の定義

プラポリ作成支援とは，表 4 に示す SPI の 8 項目のうち，②取得される情報の項目，④利用目的の特定・明示，⑥-1 外部送信・第三者提供の有無，および⑥-2 情報収集モジュールの有無，に関して，正確性と明確性を備えたプラポリの作成に有効な情報を提供することと定義する．

4.2 プログラムの静的解析によるプラポリ作成支援

プラポリ作成支援を実現する手段として，アプリが利用する API と外部モジュールを検知する静的解析を行い，解析結果に基づいてプラポリ作成に有効な支援情報を提供する機能を設計する．

SPI の 8 項目のうち，「②取得される情報の項目」は，具体的には，表 1 に示したスマホにおける利用者情報等が該当する．これら情報は，アプリが，情報を取得する所定の API（以下，情報取得 API）を実行して取得する．したがって，解析対象のプログラム中に実装された情報取得 API を検知することで，プラポリへの記載が必要と考えられる情報の提示が可能となる．

SPI の 8 項目における「⑥-2 情報収集モジュールの有無」に関する正確な記述の支援には，利用する外部モジュールの検知が有効である．また，外部モジュールについては，以下に関して，API 検知よりも多くの支援情報を提供することが可能．まず，外部モジュールは，広告表示や利用統計等の特定のサービスの利用を目的として提供されていることから，「④利用目的の特定・明示」に関する支援情報の提供も可能となる．さらに，ライブラリ化された共通のプログラムを利用するため，ある外部モジュールが利用する API や情報の送信先を事前に把握しておくことで，外部モジュールの利用を検知した場合に，「②取得される情報の項目」および「⑥-1 外部送信・第三者提供の有無」に関する支援情報の提示が可能となる．

4.3 プラポリ作成支援ツールの基本機能構成

アプリの静的解析による API と外部モジュールの検知を主たる機能とする，プラポリ作成支援ツールの基本的な機能構成を図 3 に示す．提案は，解析エンジンを主要な構成要素として，API と外部モジュールの検知に用いるデータベース（以下，DB）を備える．また，検知結果に基づいて，作成支援情報として提示するインタフェースを備える．

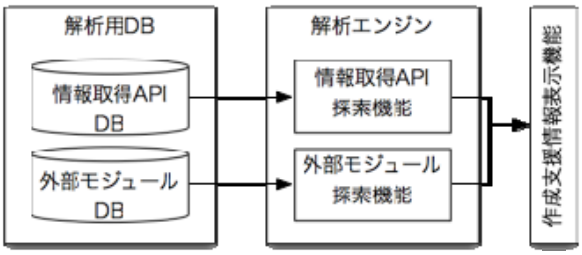

図 3 プラポリ作成支援ツールの基本的な機能構成

4.4 アプリの開発工程を考慮した実装形態の検討

プラポリ作成支援ツールは，アプリ開発環境向けのプラグインと，スマホアプリの 2 つの実装形態をとる．

開発環境向けのプラグインは，開発の上流工程に位置する利用者向けに提供する．アプリ開発に用いる開発環境と親和性が高く，プログラムのソースコードを参照できることを特徴とする．

スマホアプリは，開発の下流工程に位置する利用者向けに提供する．公開用にビルドされたアプリを容易に参照できることを特徴とする．

5. プラポリ作成支援ツールの実装

本研究では，Android アプリケーションを対象として，プラポリ作成支援ツールのプロトタイプを実装する．以下，情報取得 API DB と外部モジュール DB の実装，およびプラポリ作成支援用の Eclipse プラグインと Android アプリの実装について説明する．

5.1 情報取得 API DB の内容

情報取得 API DB は，1)アプリによる情報取得の注目対象，2)情報を取得するために利用する API，3)API を利用するために宣言が必要なパーミッション，を登録する．

我々は，注目対象として 15 種類の情報を登録した．これら情報を取得する API は 176 となり，176 の各 API の利用に宣言が必要なパーミッションの総数は 24 となった．注目情報とパーミッションの一覧を表 5 に整理する．

5.2 外部モジュール DB の内容

外部モジュール DB には，モジュール探索に必要な情報，モジュールが取得する利用者情報，モジュールの利用目的，モジュールが情報を送信する宛先を記録する．モジュール探索に必要な情報として，ライブラリのファイル名称と，クラス名称を登録する．これは，Eclipse プラグインでは，アプリをビルドする際に参照するディレクトリに追加されているライブラリのファイル名を探索の対象として，Android アプリでは，Android アプリの実行ファイルである DEX（Dalvik Executable Format）ファイルを解析して得られるクラスの名称を探索の対象とするためである．

プロトタイプにおいては，合計 131 のモジュール情報を登録した．登録したモジュールの種別に関する内訳は，広告が 99 件，利用解析が 19 件，および外部アプリ 12 件である．外部アプリとは，Facebook，Dropbox，Evernote 等の SNS やクラウドサービスと連携する機能を追加するモジュールである．

表 5 API DB に登録する各情報

注目対象	Android ID，AuthToken，Google アカウント，ICCID，IMEI，IMSI，MAC アドレス，cookie もしくはアプリ独自 ID，位置情報，電話番号，カレンダー情報，端末のシステムログ情報，インストール済みのアプリ情報，画像・音声・動画等のコンテンツ情報，電話帳情報
パーミッション (android.permission. を省略)	ACCESS_COARSE_LOCATION ACCESS_FINE_LOCATION ACCESS_WIFI_STATE ACCOUNT_MANAGER AUTHENTICATE_ACCOUNTS BLUETOOTH BLUETOOTH_ADMIN BROADCAST_PACKAGE_REMOVED CALL_PHONE DUMP GET_ACCOUNTS GET_TASKS GLOBAL_SEARCH_CONTROL PROCESS_OUTGOING_CALLS READ_CALENDAR READ_CONTACTS READ_LOGS READ_PHONE_STATE READ_SMS READ_USER_DICTIONARY RECORD_AUDIO USE_CREDENTIALS READ_HISTORY_BOOKMARKS

5.3 プラポリ作成支援 Eclipse プラグイン

Android アプリの開発には，統合開発環境である Eclipse[16]に，Android Development Tool[17]と呼ばれるプラグインを組み込んだ環境が広く用いられている．そこで，既存の開発ツールとの親和性を考慮して，プラポリ作成支援プラグインも，Eclipse のプラグインとして開発する．

5.3.1 Eclipse プラグインの概要

本プラグインは，図 3 の基本構成に従って，DB，解析ロジックおよび UI を備える．また，[15]で提案した XML 作成支援機能も備える．

5.3.2 Java のソースコードに注目した利用 API 検知

Java のソースコードを解析対象として，アプリが利用する API を検知する手順を以下に示す．

１）AndroidManifest.xml に宣言されているパーミッション情報にもとづき，情報取得 API DB に登録されている API から，宣言されたパーミッションによって利用可能となる API を選定する．

２）手順１）において選定された API を検索キーワードとして，Java のソースコードを探索する．検索キーワードに一致する文字列を発見した場合，当該の Java のファイル名と行番号を記録する．

5.3.3 ビルド・パスに追加された外部ライブラリに注目した外部モジュール検知

ビルド・パスとは，Eclipse を用いた開発において，外部モジュールを利用する際にファイルの保存先となる場所を示す．以下に，ビルド・パスを参照してアプリが利用する外部モジュールを検知する手順を示す．

1) 開発中のプロジェクトのビルド・パス情報を取得し，探索先を決定する
2) 外部モジュール DB に登録されている外部モジュールファイルの名称を検索キーワードとして，ビルド・パスに追加されているファイルを探索する．検索キーワードとファイル名が一致するものを検知した場合，DB に登録されている外部モジュールを利用していると判定する．

5.3.4 プラポリ作成支援情報の提示

5.3.2 および 5.3.3 で説明したロジックに基づく API と外部モジュールの利用に関する探索結果は，作成支援情報として UI を通じて提示する．Eclipse プラグインの作成支援情報の提示の様子として，図 4 にプラグインの UI を，図 5 に API に関する支援情報の提示を，図 6 に外部モジュールに関する支援情報の提示の様子をそれぞれ示す．図 5 および図 6 の内容は，図 4 中の「支援情報表示エリア」に表示される．

API に関する情報は，API 情報，取得する情報の項目，API を検知したクラス名，および行番号を表示する．外部モジュールに関する情報は，外部モジュールの名称，外部モジュールの利用目的，情報の送信先，およびモジュールが取得する情報の項目を表示する．

5.4 プラポリ作成支援 Android アプリ

プラポリ作成支援機能を持つ Android アプリは，アプリをインストールした端末上で，他のアプリを対象として，API と外部モジュールの利用状況を検査し，プラポリ作成支援情報を提示する機能を持つ．

5.4.1 アプリの機能構成

本アプリは，図 3 の基本構成に従って，DB，解析ロジック，UI を備える．

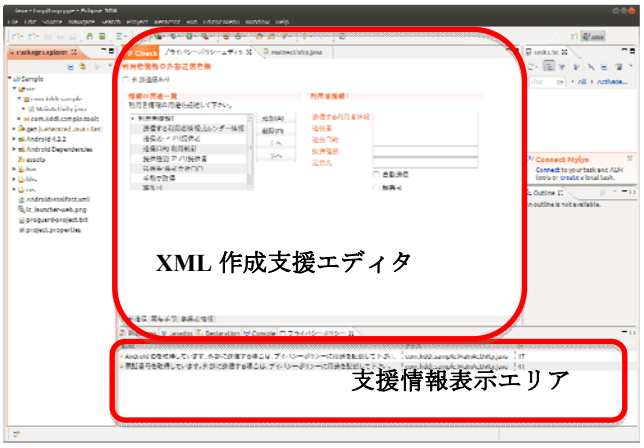

図 4 プラグインの UI

API名称	情報	クラス名	行番号
getLine1Number()	電話番号	jp.kddilabs.MainActivity.java	48
…	…	…	…

図 5 API に関する支援情報の提示

外部モジュール名称	目的	情報の送信先	取得情報
SampleAd	広告配信のため	サンプル広告株式会社	IMEI、位置情報
…	…	…	…

図 6 外部モジュールに関する支援情報の提示

5.4.2 利用 API と外部モジュールの検知ロジック

解析ロジック部は，図 7 に示す Android アプリの実行コードである DEX ファイルの構造[18]にもとづいて，API と外部モジュールを検知するロジックを作成した．以下に，解析手順を述べる．

1) AndroidManifest.xml に宣言されているパーミッション情報に基づき，情報取得 API DB に登録されている API から，宣言されたパーミッションによって利用可能となる API を選定する．
2) 手順 1)で選定した API のメソッド名を検索キーワードとして，DEX ファイルの method identifier list 領域（図 7 の method_ids 領域）を探索する．method identifier list 領域には，Java のクラス名，メソッド名，メソッド型に関する情報と，DEX ファイル中で用いる ID 情報が記録されている．検索キーワードと一致する情報が確認された場合，アプリで検索対象の API が利用されていると判定し，対応する ID 情報を保持する．
3) 手順 2)で利用していることが確認された API について，ID 情報を検索キーとして，DEX ファイルの data 領域（図 7 中の data 領域）に記録されているクラスの実装情報を探索する．ここで，data 領域には，クラス定義のほか，文字列情報やデバッグ情報も記録されているため，Class definitions list 領域（図 7 中の

class_defs 領域）を参照して，クラスの実装情報が記録されている領域を示すオフセット情報を取得する．クラスの実装情報中に，検索キーワードとして用いている ID 情報が確認された場合，検索対象の API を呼び出していることとなる．このとき，クラス名を記録する．

4) API の利用が確認されたクラス名について，外部モジュール DB に記録されているクラス名情報と照合を行う．検知されたクラス名と DB に記録されたクラス名が一致した場合，発見された API は外部モジュールによる実装であると判定する．一方，両情報が一致しない場合は，アプリ本体による実装または未知のモジュールによる実装と判定する．

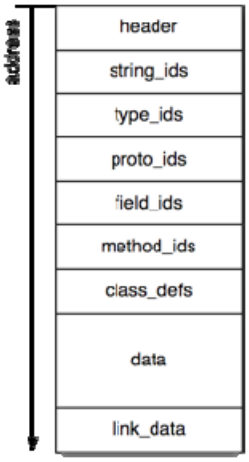

図 7 DEX フォーマットファイルのレイアウト

5.4.3 作成支援情報の提示

解析ロジックに検知した結果は UI に表示する．検査結果の示し方を図 8 に示す．表示では，検知結果について，API の実装を確認したクラス名，アプリが取得する情報の名称，実装を確認した API 名称を列挙する．視認性を高めるため，それぞれの情報の先頭には，クラス名（Class name）を示す「C:」，情報（Data）を示す「D:」，API を示す「A:」を表示する．

API が検知されたクラスについて，「アプリ本体・その他のモジュール」欄と，「外部モジュール」欄に分けて表示を行い，外部モジュールについては，モジュール名称を提示する．ここで，外部モジュール DB に登録がないモジュールが組み込まれていた場合，外部モジュールに関する情報も「アプリ本体・その他のモジュール」欄に表示される．しかし，検索結果にはクラス名称が含まれることから，クラス名称をもとに，API の検知場所がアプリ本体か，外部モジュールかの判断を行うことは容易である．

なお，一部のアプリにおいて，クラス名が「a.b.c.d」のような形式で報告されるものがあった．これは，ProGuard[19]や DexGuard[20]等の DEX ファイルの難読化ツールを用いた場合に生じることを確認した．

図 8 プラポリ作成支援 Android アプリの UI

6. 考察

6.1 各種ツールの関係性の整理と運用フレームワークの提案

SPI の 8 項目を満たし，内容に正確性と明確性を備えたアプリプラポリの管理・運用を実現するために，本研究で提案する Eclipse プラグインおよび Android アプリに，[15]で提案した XML 文書作成支援アプリを含めた運用フレームワークを図 9 に示す．

ツールの機能的側面に注目した場合，API と外部モジュール検知機能および XML 作成支援機能が必要となる．Eclipse プラグインは，単体で全てを備えているが，Android アプリは，API と外部モジュール検知機能のみを備えているため，XML 作成支援機能は，[15]で提案した XML 文書作成支援ツールを組み合わせて用いることで，アプリプラポリ作成に必要な機能を満たすことが可能となる．

アプリの開発工程に注目した場合，Eclipse プラグインは，プログラム開発者にとって親和性の高いツールとなる．一方，Android アプリと XML 作成支援ツールは，公開に向けてビルドされたファイルを対象に容易に利用できることを特徴とする．

Eclipse プラグインと Android アプリの解析精度に注目した場合，Eclipse プラグインは、Java のソースコードとビルド・パスに展開されるライブラリを直接参照することが可能であることから，解析精度が高い．一方，Android アプリ＋作成支援ツールは，APK ファイルを解析対象とすることから，難読化などを施された場合，モジュールの判断が一部困難となる可能性がある．

図 9 ツールの関係性と運用フレームワーク

6.2 期待される効果

提案した支援ツールは，SPI の 8 項目を満たす適切なプラポリの作成を支援する情報を的確に提示する．本ツールを用いることにより，2 節で説明した調査において，SPI の 8 項目をすべて満たすアプリプラポリは 4%，うち，適切な内容を備えたものは 0%という状況を改善し，SPI の 8 項目を満たした適切な内容のアプリプラポリが 100%となる状況の実現に寄与する．

7. おわりに

本研究では，アプリプラポリの正確性と明確性に関する課題を整理し，アプリの静的解析によるプラポリ作成支援機構を提案した．提案機構は，プログラムを解析して API と外部モジュールを検知し, SPI が推奨する 8 項目のうち，利用者情報の取得と外部送信に関する正確な情報に有効な情報を提供する．アプリの開発における上流工程で利用するツールとして Eclipse プラグインを実装し, 下流工程で利用するツールとして Android アプリを実装し，プラポリ作成に有益な情報を提示することを示した．提案のツールを用いることで，アプリ開発者と利用者の透明性確保に適切なアプリプラポリの作成の推進が期待される．

謝辞 本研究の 2 章の実態調査を纏めるにあたり，読み解き結果を共有頂けた日本総研(株) 東氏，前田氏に深謝する．また，アプリプラポリと事業者プラポリの区分をアドバイス頂けた産業総合研究所 高木氏，渡辺氏に深謝する．

参考文献

1) 総務省,“スマートフォン プライバシー イニシアティブ”, http://www.soumu.go.jp/menu_news/s-news/01kiban08_02000087.html http://www.soumu.go.jp/main_content/000171224.pdf
2) MCF,“スマートフォンのアプリケーション・プライバシーポリシーに関するガイドライン”, http://www.mcf.or.jp/temp/sppv/mcf_spappp_guidline.pdf
3) European Commission, April 2010, “A comprehensive approach on personal data protection in the European Union: Right to be Forgotten,” COM(2010) 609 final, Brussels. http://ec.europa.eu/justice/news/consulting_public/0006/com_2010_609_en.pdf.
4) Federal Trade Commission (FTC), 2010, “Endorses ‘Do Not Track’ to facilitate consumer choice about online tracking,” http://www.ftc.gov/opa/2010/12/privacyreport.shtm.
5) スマートフォンの利用者情報等に関する連絡協議会(SPSC), http://www.mcf.to/press_comment/pdf/spsc_press_20121004.pdf
6) 一瀬，高木，山口，渡辺，“スマホアプリにおけるアプリケーション・プライバシーポリシー掲載の現状調査”, 情処, CSEC62-61, 2013 年 7 月．
7) FLURRY, http://www.flurry.com/.
8) Google Analytics, http://www.google.com/analytics/.
9) AdLantis, http://sp.www.adlantis.jp/.
10) i-mobile, http://i-mobile.co.jp/en/index.aspx.
11) nend, http://nend.net/.
12) Admob by Google, http://www.google.com/ads/admob/.
13) AMoAd, http://www.amoad.com/.
14) 竹森敬祐，他 5 名，“スマートフォンアプリ向け独自 ID の生成・管理”，情処，CSEC59-8，2012 年 12 月．
15) 磯原, 川端, 竹森, 窪田,“XML を用いたモバイルアプリのプライバシーポリシーの運用”，信学，SCIS シンポジウム，2013 年 1 月．
16) Eclipse, http://www.eclipse.org
17) Android Development Tools, http://developer.android.com/tools/sdk/eclipse-adt.html
18) Dalvik Executable Format, http://source.android.com/tech/dalvik/dex-format.html
19) ProGuard, http://proguard.sourceforge.net
20) DexGuard, http://www.saikoa.com/dexguard