Vol.2016-CSEC-73 No.12
Vol.2016-IOT-33 No.12
2016/5/26

プログラム変換による個人情報保護の実現に向けたフレームワークのプロトタイプ実装

松永 崇秀[1,a)]　八木 良太[1]　高橋 健一[1]　川村 尚生[1]　菅原 一孔[1]

概要：
近年，インターネットの普及に伴い，オンラインショップや施設の予約など様々なネットワークサービスが利用されている．これらのサービスのいくつかは利用者に対して個人情報の提供を要求する．しかし，利用者は提供した個人情報が実際にどのように利用されるか知ることができないため，サービス提供者へ個人情報を提供することに不安を感じる．そこで，我々は利用者が個人情報の利用方法を指定することができる仕組みを提案している．本稿では，この仕組みを実現するためのプロトタイプを実装し，これを通じてプログラム変換が可能であることを確認した．

キーワード：個人情報保護，プロトタイプ，プログラム変換

Prototype Implementation of the Personal Information Protection Framework

Takahide Matsunaga[1,a)]　Ryouta Yagi[1]　Kenichi Takahashi[1]　Takao Kawamura[1]
Kazunori Sugahara[1]

Abstract: Nowadays, various network service, such as online shops and reservation of facilities, have been used with the spread of the Internet. Some of these services request to offer personal information to users. However, we cannot know how offered personal information is used. Thus, we feels uneasy to offer personal information to service providers. For this reason, we are proposing a framework that an user can designate usage procedures of his/her personal information. In this paper, we implement the prototype of our framework.

Keywords: personal information protection, prototype, program conversion

1. はじめに

近年，インターネットの普及に伴い様々なネットワークサービスが利用されている．例えば，Amazon や楽天などのオンラインショップや Gmail などのメールサービス，ホテルの予約やオンラインバンキングなどが挙げられる．これらのサービスは，利用者に対して名前や住所，電話番号やクレジットカード番号などの個人情報の提供を求める．利用者は要求に従い自身の個人情報を提供することでサービスを利用することができる．しかし，サービス提供者の要求する情報の種類，提供の方法に従わなければ，利用者はサービスを利用することができない．例えば，多くのサービス提供者では情報の登録時において入力必須欄が存在しており，その項目の情報については提供しなければならない．これらの情報は HTTPS などにより暗号化してサービス提供者に送信されることが多い．しかし，このような暗号化が行われていなかった場合，たとえ利用者が暗号化することを望んだとしても暗号化することはできない．また，利用者には一度サービス提供者に提供した個人情報が実際にどのように利用されているか確認する術がない．こ

[1] 鳥取大学大学院工学研究科
Graduate School of Engineering, Tottori University
[a)] s112052@ike.tottori-u.ac.jp

のため，利用者はサービス提供者に個人情報を提供することに不安を感じたとしても，個人情報を提供してサービスを利用するか，個人情報を提供せずにサービスを利用しないかという選択しかすることができない．

そこで，我々は利用者自身が個人情報の処理方法を決めることができる仕組みを提案している [1], [2]．利用者が提供した個人情報はサービス提供者が持つプログラムで処理される．そこで，このプログラムの処理方法を利用者が指定した方法に書き換える．これにより，利用者が指定した方法でサービス提供者に個人情報の処理を行わせる．すなわち，個人情報を提供する利用者自身が個人情報の利用方法を決定するため，利用者は安心して個人情報を提供することができる．本稿では，この仕組みを実現するためのプログラム変換およびフレームワークの実装について述べる．

2. 関連研究

提供する個人情報の利用目的や利用方法などを記したプライバシーポリシ [3] を多くのサイトが策定している．利用者はプライバシーポリシを閲覧することにより，サービス提供者による個人情報の利用方法を知ることができる．しかし，プライバシーポリシはサービス提供者によって記述された通りに情報を利用することを保証するものではない．また，多くの利用者がプライバシーポリシを閲覧しないという問題も存在する．そこで，収集する個人情報の利用方法を利用者に提示するフォーマットとして P3P（Platform for Privacy Preferences）[4] が提案されている．P3P は利用者があらかじめ定めた個人情報の利用基準と各サイトのプライバシーポリシを比較し，自動的に情報提供の可否を判断する．しかし，P3P もサービス提供者がプライバシーポリシ通りに情報を利用することを保証しない．

また，個人情報を送信しないことにより，悪意のあるアプリケーションから個人情報を守ること [5] が提案されている．この研究では，個人情報の代わりにアプリケーションルールにより生成された制御コマンドを送信することで情報を保護する．しかし，アプリケーションルールはサービス提供者が作成しているため，サービス提供者のことを信頼出来ない利用者の不安は解消することができない．

利用者が安心してサービスを利用するための仕組みとして，PPM（Privacy Policy Manager）[6] が提案されている．PPM では，パーソナルデータの取り扱いに関するユーザプリファレンスを管理することによってデータの流通を制御する．また，開示する情報の粒度を制御することでプライバシを保護する研究 [7], [8] が行われている．これらにより，利用者の要望にあった情報のみを送信し，その情報のみで利用できる範囲のサービスを利用することが可能となる．しかし，その情報単体で利用者の特定に繋がる情報は守ることができない．また，サービス提供者が必要とする情報を遮断すると，サービスの利用に支障を来す可能性がある．

3. 個人情報保護フレームワーク

本フレームワークでは，サービス提供者の持つ個人情報処理プログラムに利用者が指定した処理方法を適用することで，利用者の意図を個人情報の処理に反映させる．これを実現するためには，利用者は自身の安心できる処理方法をサービス提供者に伝える必要がある．

ここで，サービス提供者による個人情報の処理方法はプログラムによって決まっている．このため，利用者が安心できる処理方法は，このプログラムに適用可能な処理方法である必要がある．そこで，プログラム中での情報の利用方法を示した利用ポリシを定義する．利用ポリシにより，利用者は間接的にサービス提供者による個人情報の利用方法を知ることができる．

また，個人情報の処理方法およびプログラムの変換方法を定義した保護ポリシを準備する．保護ポリシは信頼できる第三者機関 (TTP：Trusted Third Party) が設置した保護ポリシデータベースで管理されるものとする．TTP はサービス提供者のプログラムを保護ポリシに従って変更するサービスも提供する．利用者は利用ポリシを参照することで，そのプログラムに適用可能である保護ポリシを選択し，サービス提供者に伝える．サービス提供者は保護ポリシに従ってプログラムを変換し，変換後のプログラムで個人情報の処理を行う．これにより，利用者が個人情報の保護方法を決定する．

3.1 システムの動作の流れ

個人情報保護フレームワークの概要を**図 1** に示す．

(1) 利用者はサービス提供者にサービスの利用を要求する．
(2) サービス提供者は利用者に対して個人情報の提供を要求するとともに利用ポリシを送信する．これにより，プログラム中での個人情報の利用方法と処理内容を利用者に伝える．
(3) 利用者は利用ポリシを参照することで，TTP からサービス提供者のプログラムに適用可能な保護ポリシの一覧を取得し，自身の安心できる保護方法が記述された保護ポリシを選択する．
(4) 利用者は自身の個人情報に保護ポリシを適用する．
(5) 利用者は保護ポリシを適用した個人情報と選択した保護ポリシをサービス提供者に送信する．
(6) サービス提供者は TTP に個人情報処理プログラムと利用ポリシ，および利用者が選択した保護ポリシを送信する．
(7) TTP はサービス提供者から受け取った個人情報処理プログラムを保護ポリシに従って変換する．
(8) TTP は変換後の個人情報処理プログラムをサービス提供者に送信する．

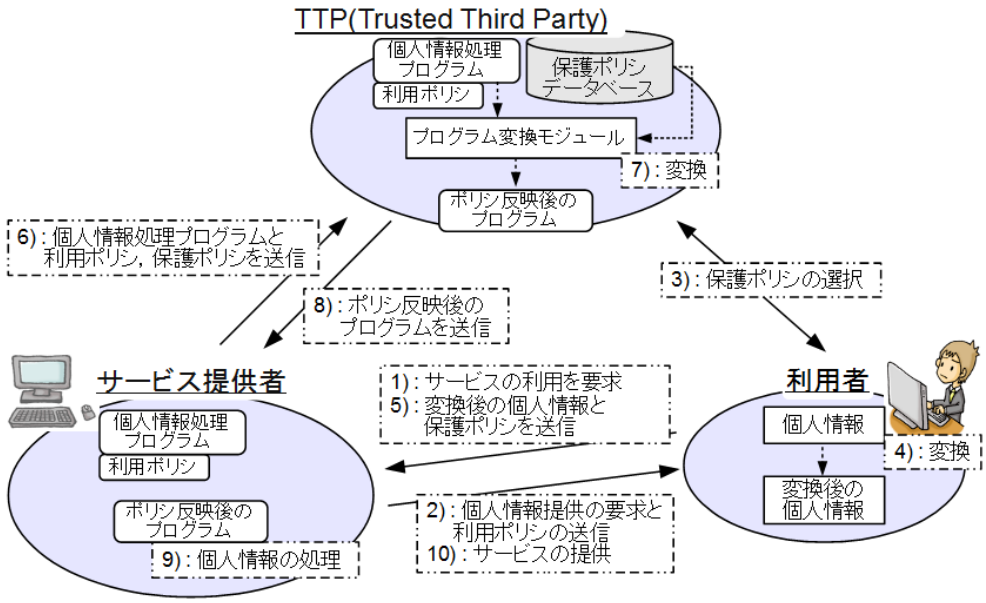

図 1 個人情報保護フレームワーク
Fig. 1 Overview of the Personal Information Protection Framework

(9) サービス提供者は変換後の個人情報処理プログラムで個人情報の処理を行う．
(10) 個人情報の処理後，サービス提供者は利用者にサービスを提供する．

本フレームワークにより，利用者は自身の選択した方法により個人情報の処理を行わせることができるため，保護方法の決定権を持つことができ，安心してネットワークサービスを利用できる．

3.2 保護ポリシと利用ポリシ

本フレームワークでは，個人情報処理プログラムを保護ポリシに従って変換することで，利用者が個人情報の保護方法を決定する．ここで，プログラムを変換するためにはプログラム中での個人情報の処理方法を知る必要がある．そこで，利用ポリシを利用する．**図 2** 左に利用ポリシの例を示す．

利用ポリシは各個人情報について準備され，個人情報処理プログラム中で個人情報がどの変数に格納され，どのように処理されるかが定義される．図 2 の例はパスワードに関する利用ポリシを示している．7 行目にはパスワードが利用ポリシ中で “password” という識別子で表現されていることが示されている．また，9 行目はパスワードを password.equals(spPass) という形式で利用することを示している．12〜15 行目から，“pass” という名前がついた入力フォームに入力されたデータがパスワードを表し，プログラム中で strPass という変数で利用されていることがわかる．同様に，17〜20 行目を見ると spPass はプログラム中で registeredPass 変数で利用され，getPassword(id) によって初期化されていることがわかる．

また，個人情報の保護方法およびプログラムの変換方法は保護ポリシとして定義する．パスワードをハッシュ化することで保護するポリシの例を図 2 右に示す．7〜10 行目には，パスワードが保護ポリシ中で “p1” という識別子で表現され，p1.equals(p2) という形式で利用されるときに適用可能であることが示されている．12〜14 行目には，p1.equals(p2) でのパスワードの利用を，hashedUserPass.equals(hashedSpPass) に変換しなければならないことが示されている．さらに，hashedUserPass は p1 を md5HashGen メソッドによって変換して生成すること（16〜21 行目）や，hashedSpPass は p2 を md5HashGen メソッドによって変換して生成すること（23〜28 行目）が示されている．

これらのポリシを結びつけることで，プログラムに適用可能な変換ルールを導出する．保護ポリシは利用ポリシを参照した利用者によって選択されているため，2 つのポリシは同じ情報に対するポリシであることがわかる．そこで，これらのポリシの<OBJECT>を参照し，ポリシの結びつけを行う．図 2 右の保護ポリシが図 2 左の利用ポリシに適用するものとして選択されたとすると，利用ポリシ中の password は保護ポリシ中の p1 に対応することがわかる．同様に，利用ポリシ中の spPass は保護ポリシ中の p2 に対応することがわかる．さらに，password はプログラム中で strPass 変数として，spPass は registeredPass 変数として利用されていることが利用ポリシからわかるため，保護ポリシ中の p1.equals(p2) は strPass.equals(registeredPass) で実現されていることがわかる．これにより，strPass.equals(registeredPass) を hashedUserPass.equals(hashedSpPass) に変換するための変換ルールを導出する（ルール 1）．また，strPass は入力

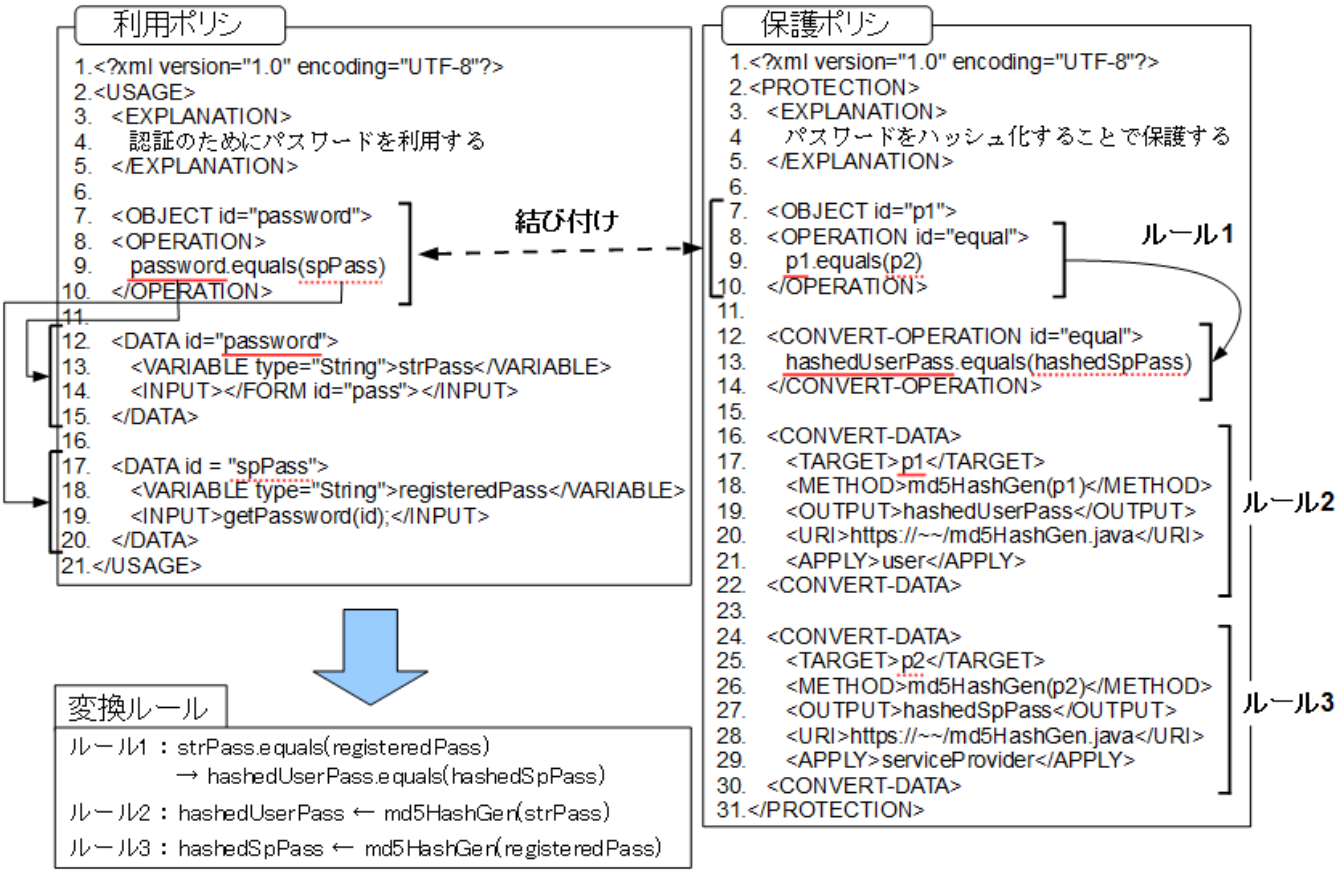

図 2 利用ポリシと保護ポリシの例
Fig. 2 Examples of an use policy and protection policy

フォームに入力されたデータであることから，このデータに対して md5HashGen メソッドを適用し，hashedUserPass を生成する変換ルールを導出する（ルール 2）．同様に，hashedSpPass は getPassword(id) で得られる値に対して md5HashGen メソッドを適用し生成する（ルール 3）．これにより，保護ポリシが保護対象とする情報が，プログラム中でどの変数に格納され，どのような操作で処理されるか知ることができる．

3.3 プログラムの変換

変換ルールの導出後，プログラムを変換する．プログラムを変換するためには変換ルールを適用すべき場所を特定する必要がある．そこで，プログラムの解析を行う．ここでは，Java 言語で記述されたプログラムを対象とし，Java 言語のソースコードを解析することができる ASTParser[9] を利用する．ASTParser は EclipseJDT が提供する API の 1 つで，Java 言語のソースコードを解析し，抽象構文木を生成する．抽象構文木とは，ソースコードからコメント文や空行などの実行する際に不要な情報を取り除いたデータ構造のことである．ASTParser により構文解析され，プログラムの各行は変数の宣言文や代入文，ループ文などのように意味付けされる．またその行はさらに字句へと分解され，変数名やメソッド名などのように意味付けされる．これにより，プログラム中でどの変数にどの情報が格納されているか，どのメソッドでどの変数が使用されているかが解析できる．解析後，変換ルールを変換対象のプログラムに適用し，プログラムを変換する．すべての変換ルールを適用することで，利用者が選択した保護方法を適用した個人情報処理プログラムを生成する．

4. フレームワークの実装

本フレームワークの有効性を確認するために，サービス提供者，利用者，TTP のプロトタイプを実装する．

4.1 サービス提供者の実装

サービス提供者の機能として

- 利用ポリシの通知
- 保護ポリシと保護ポリシ適用後の個人情報の受信
- 変換後の個人情報処理プログラムの実行

が必要となる．

サービス提供者は利用者からサービス利用の要求があった場合，利用者に対して利用ポリシを通知する必要がある．利用者への利用ポリシの通知は，個人情報入力のための Web ページのヘッダ部分に，そのページで入力を求め

```
postContent = getPostContent();     //利用者からのPOSTデータ
pPolicy = getPPolicy(postContent);  //保護ポリシの参照先
convInfo = getCInfo(postContent);   //変換後の個人情報

s = new SSLSocket(TTP_addr, TTP_port); //TTPに接続
s.send(pPolicy, uPolicy, myProgram);       //TTPに送信

convProgram = s.receive();  //変換後のプログラムを受信
// 受信したプログラムをコンパイル
new ProcessBuilder("javac", convProgram).start();
// プログラムを実行
new ProcessBuilder("java", convProgram, convInfo).start();
```

図 3 サービス提供者のプログラムの概要
Fig. 3 Overview of program in service provider

る個人情報に対する利用ポリシの URI を記述しておくことで実現する．

保護ポリシおよび保護ポリシ適用後の個人情報の受信は HTTP POST で実現する．POST された内容から保護ポリシの参照先（URI）と保護ポリシ適用後の個人情報を取得する．これらの情報をローカルに保存した後，TTP が提供するプログラムの変換サービスを利用するために，TTP へ個人情報処理プログラムと利用ポリシ，保護ポリシの参照先を送信する．これらの情報の送信は Java で実装した SSL ソケット通信により実現する．

その後，サービス提供者は TTP から変換後の個人情報処理プログラムが送られてくるのを待つ．受信後，プログラムをローカルに保存し，保存した場所を参照して受け取った個人情報処理プログラムをコンパイルし実行する．これにより，変換後の個人情報処理プログラムで保護ポリシ適用後の個人情報を処理する．これらの処理を行うためのプログラムの概要を**図 3** に示す．

4.2 TTP の実装

TTP は

- 保護方法選択ページ
- 個人情報処理プログラムの変換

の機能を提供する．

TTP が提供する保護方法選択ページの例を**図 4** に示す．図 4 の左上のフレームには利用者側のアドオン（後述）により解析された利用ポリシの一覧が表示される．すなわち，利用者が個人情報入力画面で要求されている個人情報の一覧が表示される．図 4 はサービス利用時に ID とパスワードを要求するサービスに対する例であることを示す．ここで，ID/Password の横に配置された選択ボタンが押されると，選択された個人情報に対して適用可能な保護ポリシの一覧が左下のフレームに表示される．保護ポリシは TTP 内の保護ポリシデータベースで管理されており，MySQL で実装されている．一覧の中の保護ポリシをクリックすると，その保護ポリシが個人情報をどのように保護するものであるかが右下のフレームに表示される．利用者はこれを

図 4 保護方法選択のためのユーザインターフェース
Fig. 4 User interface for selecting a protection method

見ることで，自身の安心できる保護方法が記述された保護ポリシを選択する．右下のフレームの決定ボタンが押されると，左上のフレームに選択した保護ポリシが記録される．保護ポリシ選択後，利用者が決定ボタンを押すことで，保護ポリシが利用者に送信される．

また，TTP はプログラム変換モジュールにより個人情報処理プログラムを変換するサービスを提供する．TTP は個人情報処理プログラムと利用ポリシ，保護ポリシの参照先を受け取ると，プログラム変換モジュールにこれらの情報を与え，個人情報処理プログラムを変換する．プログラムの変換後，サービス提供者にプログラムを返す．

4.3 利用者の実装

利用者には

- 利用ポリシの取得および保護ポリシの選択
- 個人情報への保護ポリシの適用と情報の送信

を実現するための機能が必要となる．

これらは Google Chrome のアドオンとして実装する．サービス提供者の個人情報入力ページで右クリックをしてアドオンを使用すると，個人情報入力ページのヘッダに示された URI から利用ポリシを取得し，TTP の保護方法選択ページ（図 4）へアクセスする．利用者は 4.2 節で述べた手順で保護ポリシを選択，取得し，ローカルに保存する．

利用者が個人情報入力ページで個人情報を入力し送信ボタンを押すと，入力した個人情報は取得した保護ポリシに従って変換される．これにより利用者の個人情報に保護ポリシが適用され，変換後の個人情報を生成する．その後，選択した保護ポリシの参照先と変換後の個人情報をサービス提供者に送信する．

4.4 実験

個人情報の変換処理を正しく行うことができるかどうかを実験により確認した．サービス提供者が生のパスワード

図 5 パスワードの認証結果
Fig. 5 The result of password authentication

を受け取り，パスワード認証に成功したか否かを表示するだけのサービスを想定して実験を行った．利用者がハッシュ変換によりパスワードを保護する保護ポリシを選択した結果を**図 5** に示す．

パスワードとして「java_password」と入力して送信したところ，サービス提供者にはハッシュ変換後の「54cbc・・・」が送信され，この情報により正しく認証ができていた．これにより，保護ポリシによって個人情報処理プログラムが変換され，正しくパスワードの処理ができていることが確認できた．

5. おわりに

本稿では，インターネットサービス利用時に利用者自身が個人情報の利用方法を選択することができる個人情報保護フレームワークの実装について示した．これにより，個人情報およびプログラムの変換を行い，正しく処理を行うことができたことを確認した．

参考文献

[1] Takahashi, K., Matsuzaki, T., Mine, T., Kawamura, T. and Sugahara, K.: Protection of Personal Information based on User Preference, *IJNCAA*, Vol.1, No.4, pp.822-834 (2011).
[2] 松永崇秀，高橋健一，川村尚生，菅原一孔：個人情報保護を目的としたフレームワークの提案，コンピュータセキュリティシンポジウム 2015(CSS2015)，pp.162-169 (2015).
[3] IBM：プライバシー・ポリシーの定義，入手先 ⟨https://publib.boulder.ibm.com/tividd/td/ITPME/SC23-1284-00/ja_JA/HTML/p12plmst22.htm⟩ (2016.04.15).
[4] W3C：Platform for Privacy Preferences (P3P) Project，入手先 ⟨http://www.w3.org/P3P/⟩ (2016.04.15).
[5] 田丸修平，岩谷晶子，高汐一紀，徳田英幸：プライバシを考慮したパーソナライゼーションを実現するアプリケーションフレームワーク，情報処理学会研究報告システムソフトウェアとオペレーティング・システム（OS），pp.49-56，2003.
[6] 中村徹，Andrew A. Adams，村田潔，清本晋作，高崎晴夫，渡辺龍，三宅優：パーソナルデータ流通基盤: Privacy Policy Manager (PPM) の受容性評価，暗号と情報セキュリティシンポジウム（SCIS2014），3D3-2，2014.
[7] 宮本崇弘，竹内亨，奥田剛，春本要，有吉勇介，下條真司：プライバシとサービス品質のトレードオフを考慮した個人情報制御機構の提案，第 16 回データ工学ワークショップ（DEWS2005），6-A-01，2005.
[8] 菊池亮，高橋克巳：ログ情報活用におけるプライバシー保護技術の考察，情報の科学と技術，Vol.63，No.2，pp.69-73，(2013).
[9] Manoel Marques：Eclipse の ASTParser を試す，IBM developerWorks，入手先 ⟨https://www.ibm.com/developerworks/jp/opensource/library/os-ast/⟩ (2016.04.15)