プログラム解析基盤のサービス化

桑　原　寛　明[†1]

本稿では，プログラム解析技術の簡便な利用と連携の実現を目指すプログラム解析基盤のサービス化について議論する．

Towards Program Analysis Platform as a Service

Hiroaki Kuwabara[†1]

In this paper, we discuss a service of program analysis platform. The goal of this service is to reduce the cost to use and/or combine program analysis technologies.

1. はじめに

リエンジニアリングの出発点は多くの場合ソースコードである．そのため，プログラム解析技術を利用したソースコードのリバースエンジニアリングは，リエンジニアリングにおける重要な工程の一つである．

リバースエンジニアリングにおいて，開発者はソースコードを読んで理解し必要な情報を抽出する．その作業を支援するために，ソースコードからクラス図やシーケンス図を復元する設計復元，部品抽出，クロスリファレンス，波及解析，プログラムスライシング，抽象実行，リファクタリング，コードクローン検出，メトリクス測定などソースコードを対象とする処理を行う様々なツールが開発されている．ソースコードを処理するツールの大半は，字句解析，構文解析，意味解析，フロー解析といった基礎的な解析を必要とする．そのため，これらの解析を行う解析器とその解析結果を利用するための API を提供するプログラム解析基盤が数多く開発されており，例えば WALA，Soot，**Sapid** などがある．これらの解析基盤を用いることで個々の要求に対応したツール本来の機能の実現に専念できる．

これらのツールや解析基盤は基本的にローカルな環境で実行される．ツールや個々の解析技術がサービスとして提供されたり，解析結果がサーバ上で共有されるといったことはあまり見られない．そのため，独自のツールを実装するための敷居が高い，同じソースコードに対して同じ解析が実は何度も行われている，といった状況が発生しやすい．

本稿では，リバースエンジニアリングをうまく進めることがリエンジニアリングにおける重要なポイントの一つであるとの立場で，著者らが提案するプログラム解析技術のサービス化[1)] について述べ，提供中あるいは提供予定の具体的なサービスの例を挙げる．プログラム解析サービスの可能性と有用性，サービス化すべき機能やツールについて議論したい．

2. プログラム解析サービス

プログラム解析サービスの目的は，プログラム解析を行うリバースエンジニアリングツールの効率的な開発に必要な支援を提供することである．**図 1** に示すように，構文情報生成（字句解析，構文解析，意味解析）のような基礎的かつ汎用的な解析機能と，コーディング検査のような個々の目的に特化した機能をサービスとして提供する．ツールの開発者は，既存のサービスを組み合わせる，構文情報生成サービスやフロー解析サービスの結果を用いて独自の解析を実現する，既存サービスに独自解析を組み合わせる，といった手段で必要なプログラム解析を実現できる．さらにユーザインタフェースを構築することでツールを作成する．

現在，以下に挙げる解析機能を **Sapid** が持つ JavaScript 向けの解析系と Java Servlet を用いて Web サービスとして試作している[⋆1]．各サービスには固有の URI を持ち，HTTP を利用してアクセスできる．サービスの入力はリクエストパラメータとして与え，出力はレスポンスのボディに含められる．

†1 立命館大学情報理工学部
College of Information Science and Engineering, Ritsumeikan University

⋆1 http://sapidjsx.appspot.com/

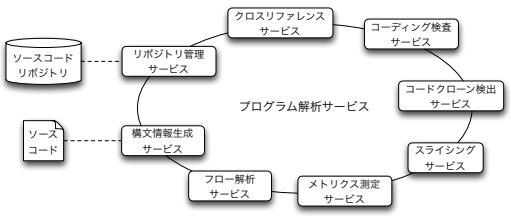

図 1 プログラム解析サービスの概要
Fig. 1 Overview of Program Analysis Service

構文情報生成サービス ソースコードに対して字句解析, 構文解析, 意味解析を行って抽象構文木と記号表に相当する構文情報を生成するサービスである．構文情報はほぼすべてのプログラム解析において必要とされる最も基本的な情報である．POST されたソースコードを **Sapid** の解析器を用いて解析し，生成された XML 形式の構文情報を返す．

リポジトリ管理サービス 構文情報やフロー解析の結果はソースコードが変わらない限り同じであるため，解析結果をリポジトリに蓄積して共有すれば何度も解析しなくてもよい．リポジトリ管理サービスはリポジトリへのアクセスを制御するサービスである．ソースコードのファイルを単位としてソースコードのハッシュに基づいて解析結果を管理する．ファイル名の扱いや複数のファイルをまたがる解析結果の管理は今後の課題である．

制御フロー解析サービス 制御フローグラフを生成し，その XML 表現を返すサービスである．生成には構文情報が必要であるため構文情報生成サービスと連携する．

コーディング検査サービス コーディング検査を行い，その結果を表す XML 文書を返すサービスである．実際の検査は **Sapid** を利用したコーディング検査器[2] による．

これらのサービスの実装と同時に，どのような機能をサービスとして提供すべきか，サービスの入出力をどのように表現するか，複数のサービスをどのように連携させるか，といった課題について検討を進めている．現状では，既存の解析機能のサービス化に留まっており，知見を得るためにサービスを利用した新しいツールの開発を行う必要があると考えている．

3. 議　　論

3.1 サービス化の利点と欠点

ツールの利用者および開発者における利点として以下の点が挙げられる．

- サーバ側で解析を行うため誰もが同じ解析機能と解析結果を共有できる
- 解析のスケーラビリティを確保する責任をサービスの提供者が持つ
- HTTP 通信と XML 操作が可能であれば任意のプログラミング言語を用いてツールを実装できる

一方，欠点として以下の点が挙げられる．

- サービスの呼び出しがネットワークを介して行われるためレイテンシや入出力のサイズが問題になる可能性がある
- ソースコードがどこまで流れていくか直感的でなく制御も困難である

後者の欠点は企業などでの利用においては特に問題であり，サービスに入力を送るのではなく，手元にサービスを持ってきて実行できるような仕組みが必要なのではないかと考えている．

3.2 リエンジニアリングにおける解析基盤

本稿では，リエンジニアリングにおける個々のプログラム解析技術に着目するのではなく，プログラム解析技術を提供しリバースエンジニアリングツールの開発を支援するための解析基盤に着目している．解析基盤のサービス化によって，解析結果の共有や解析技術あるいはツールの連携が促進されることを期待している．

リバースエンジニアリングにおいて具体的にどのような解析技術が必要とされるかはサービスの提供者にとって興味深い問題である．構文情報生成やフロー解析のような基礎的かつ汎用的な解析と，それらを利用して実現される設計復元のようなツールとの間のギャップは小さくない．通常はツールの開発者がそのギャップを埋めるが，ギャップをより小さくするために解析基盤が何をどのように提供すべきか議論したい．

リエンジニアリングにおいてはフォワードエンジニアリングも当然に重要であるが，フォワードエンジニアリングにおけるプログラム解析技術についても議論したい．リエンジニアリングが繰り返されるのであれば，フォワードエンジニアリングにおいてリバースエンジニアリングを考慮する必要があると考えられる．

参　考　文　献

1) 桑原寛明，渥美紀寿，山本晋一郎：プログラム解析技術のサービス化の試み，ソフトウェア工学の基礎 XVIII (FOSE 2011)，近代科学社，pp. 243–248 (2011).
2) 桑原寛明，末次　亮，山本晋一郎，阿草清滋：拡張可能な JavaScript 向けコーディング検査器，ソフトウェア科学会第 27 回大会，6C-1 (2010).