

スクリプト言語向け CASE ツール開発環境

桑 原 寛 明^{†1}

本稿では、JavaScript を初めとするスクリプト言語を対象とした CASE ツール開発環境が持つべきであると考えられる機能について議論する。

A CASE Tool Platform for Scripting Languages

HIROAKI KUWABARA^{†1}

In this paper, we discuss which functions CASE tool platform should provide for targeting scripting languages.

1. はじめに

我々の研究グループでは Web アプリケーションを対象とした CASE ツール開発環境の研究・開発を行っており、2 年程前から Ajax 技術を利用したアプリケーション（以下、Ajax アプリケーション）を対象に加えてきた。CASE ツール開発環境の基礎の部分には CASE ツール・プラットフォームである Sapid^{2),4)} を採用し、必要に応じて Sapid の拡張を行いながら開発環境を構築している。

Ajax 技術においては JavaScript が重要な役割を果たしており、JavaScript プログラムとその周辺をうまく解析する能力が求められている。本ワークショップでは、Ajax アプリケーション向け CASE ツールを紹介しながら、JavaScript のようなスクリプト言語を対象とする CASE ツールの開発時に CASE ツール開発環境に期待する機能について議論したい。

2. Ajax アプリケーション向け CASE ツール

Ajax アプリケーションを構成する JavaScript プログラムを対象とした CASE ツールを 2 種類紹介する。

2.1 JavaScript コーディング検査

JavaScript はインタプリタ型であるため Web ブラウザなどで実際に実行されるまでエラーの存在は分からない。また、動的型付け言語であり、仮引数と実引数の個数が異なってもよい、同名の変数や関数を複数回宣言できる、などプログラムを書く上での制限が

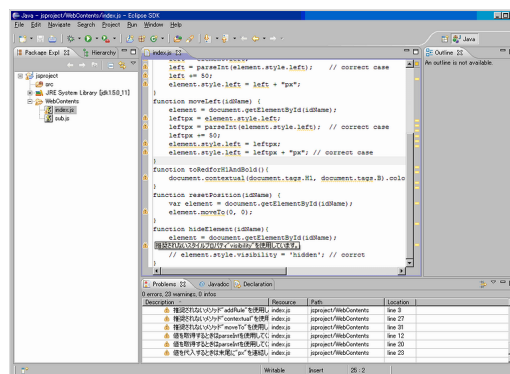


図 1 JavaScript コーディング検査器

緩い^{*1}。加えて、Ajax アプリケーションでは XMLHttpRequest や DOM、スタイルの操作を JavaScript プログラムによって行うが、その操作にはある種の作法が存在する。Web ブラウザ間の互換性も問題になりやすい。

そこで、JavaScript プログラムを静的解析技術を用いて検査するツールを開発した（図 1）。Sapid の解析器を用いて細粒度リポトリを生成し、そのリポトリに対して検査を行う。本稿の執筆時点で検査項目は 27 種類用意されている。個々の検査項目は一つの Java クラスとして実装されており、設定ファイルに記述することで任意に差し替え可能になっている。検査項目として

- 同名の変数や関数の複数回宣言
- XMLHttpRequest の open メソッドの呼び出し

^{†1} 立命館大学 情報理工学部

College of Information Science and Engineering,
Ritsumeikan University

*1 一方で、解析を難しくする要因の一つになっている

における引数の個数

- 標準ではない DOM のプロパティやスタイルの使用

などがある。

2.2 アプリケーション全体のフロー解析

Ajax アプリケーションにおいてセキュリティの問題、例えばフォームから入力された値が未加工のままデータベースに到達したり HTML に埋め込まれる可能性の検出への応用を一つの目的として、Ajax アプリケーションのデータフロー解析を開発している。Ajax アプリケーションにおいて JavaScript プログラムは Web ブラウザが描画する HTML とサーバ側プログラムの間に位置している。そのため、JavaScript プログラムのデータフロー、サーバ側プログラムのデータフロー、および各データフローや HTML との間の接続関係を明らかにすることで Ajax アプリケーション全体のデータフローを解析する。

JavaScript プログラムのデータフロー解析に関して、本稿の執筆時点ではある程度実装されているが、関数オブジェクトなど JavaScript に特徴的な要素はあまり考慮できていない。

3. CASE ツール開発環境に対する要求

構成関係や参照関係、各種フロー解析に加えて、以下のような機能も提供されることが望ましい。

3.1 型情報

JavaScript のようなスクリプト言語では動的型付けが採用されることが多い。そのため、変数や関数の型を単純にソースコードから読み取ることはできず、インスタンスを生成している箇所とデータフローの情報を組み合わせるなどして型を推測しなければならない。例えば、メソッド呼出しにおけるレシーバのクラスはすぐにはわからず、フロー解析が提供されていても推測の手間がかかる。スクリプト言語を対象とする場合、CASE ツール開発環境が型の推測も行うべきであり、変数に加えて式に対してもその値の型をうまく扱うための枠組みが必要である。

3.2 ライブラリ・フレームワーク対応

スクリプト言語に限らず一般的に言えることであるが、通常、アプリケーションの構築にはライブラリやフレームワークが利用される。例えば、Ajax アプリケーションではブラウザ互換性の問題が顕著なこともあり、prototype.js を初めとして各種存在するライブラリの利用が当たり前になっている。このような状況でプログラム解析を行う場合、ライブラリやフレームワークのソースコードが入手できなかつたり、入手で

きても解析が非常に手間であることが多い。

そこで、CASE ツール開発環境にライブラリやフレームワークに関する知識を宣言的に与え、その知識を解析に活用する仕組みが望まれる。例えば、prototype.js を利用した Ajax アプリケーションの解析において、Ajax.getTransport() が XMLHttpRequest オブジェクトを返すことを知識として与えられていれば、その内部で XMLHttpRequest オブジェクトを生成して返していることを解析して突き止めなくてもよい。

3.3 意味解析系の生成

CASE ツール開発環境においてリポジトリを生成する処理はコンパイラの処理と類似している。そのため、字句解析や構文解析には Lex や Yacc のようなツールが活用できる。参照関係を明らかにするためにコンパイラと同様に意味解析も行う必要があるが、適当なツールが存在しない。最近の CASE ツール開発環境は複数のプログラミング言語を扱う必要があり、各言語の意味解析の実装には多大な労力が必要である。

記号表ライブラリや意味解析器の生成に関する研究はいくつか行われている^{1),3),5)-7)}。しかし、様々なスコールールや動的な型に対応しながら宣言的に意味解析を記述する手法についてさらに研究を進める必要がある。

参 考 文 献

- 1) Pei-Chi Wu, Jin-Hue Lin, and Feng-Jian Wang. Designing a Reusable Symbol Table Library. In TENCON'94, pages 892-896, 1994.
- 2) 吉田 一, 山本 晋一郎, 阿草 清滋. XML を用いた汎用的な細粒度ソフトウェアリポジトリの実装. 情報処理学会論文誌, 44(6):1509-1516, 2003.
- 3) 佐々 政孝, 石塚 治志, 中田 育男. 1パス型属性文法に基づくコンパイラ生成系 Rie. コンピュータソフトウェア, 10(3):20-36, 1993.
- 4) 福安 直樹, 山本 晋一郎, 阿草 清滋. 細粒度リポジトリに基づいた CASE ツール・プラットフォーム Sapid. 情報処理学会論文誌, 39(6):1990-1998, 1998.
- 5) 高橋 透, 大久保 弘崇, 粕谷 英人, 山本 晋一郎. XML を用いた CASE ツール・プラットフォーム作成支援環境. 日本ソフトウェア科学会 FOSE'04, pages 161-172, 2004.
- 6) 亀山 裕亮, 中井 央, 山下 義行, 田中 二郎. コンパイラのための意味解析器生成系. 日本ソフトウェア科学会第 18 回大会, 2001.
- 7) 舞田 純一, 中井 央. コンパイラにおける意味解析器の自動生成. 情報処理学会論文誌: プログラミング 46(SIG14):67, 2005.