動的解析を用いるCASEツールプラットフォームの情報取得について

堀田 吉彦*，大久保 弘崇，粕谷 英人，山本 晋一郎 (愛知県立大学), 斉藤 邦彦 (滋賀大学)

On Information Acquisition of CASE Tool Platform using Dynamic Analysis

Yoshihiko Hotta, Hirotaka Ohkubo, Hideto Kasuya, Shinichiro Yamamoto (Aichi Prefectural University), Kunihiko Saito(Shiga University)

1. はじめに

我々は，プログラムの動作を CASE ツールプラットフォームで統一的に扱う為の振舞いモデルを提案している．振舞いモデルは言語独立であり現在は言語と Java に対応している．本稿では，Java プログラムからの動的情報の取得と振舞いモデルの構築について述べる．

2. 動的解析ツールの情報取得

動的解析プラットフォームの情報取得部は Java 仮想機械から直接情報を取得する JDI を用いる．先行してソースプログラムを静的解析することにより，必要な情報取得部を特定し，JDI のみでは困難な振舞いの取得を行なう．

3. 動的解析 CASE ツールプラットフォーム

<3 · 1> プログラムの振舞いモデル

プログラムの振舞いモデルは，抽象動作の実行系列，記憶領域，値の三つの要素で構成される．抽象動作の実行系列は，プログラムの振舞いの抽象的な動作を 14 種の抽象動作に分類する．記憶領域は，値を格納するためのメモリの領域をモデル化したものである．値は，整数，浮動小数点数，文字などの基本型の値，ポインター，参照などの参照型の値をモデル化したものである．この三つの要素が，関連によって関係付けられている．動的解析情報の取得では，プログラムの実行から，これらのモデル要素や関連を抽出し振舞いモデルの構築を行う．

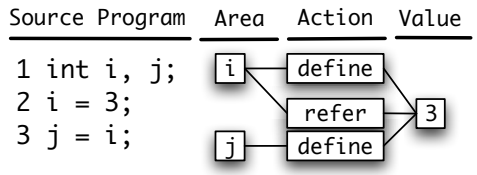

図 1 振舞いモデルの例

図 1 は，図中左部のソースプログラムの動作を振舞いモデルで表現している．中央から右に向って領域，抽象動作列，値の順に並んでいる．矩形はモデルの要素を表し，矩形間の線は要素間の関連を表す．

<3 · 2> 動的解析情報の取得部

Java プログラムの動作情報の取得には JDI/JVMTI を用いる．JDI は JVMTI から，対象 VM で発生した情報をイベント形式で取得することが可能である．

抽象動作の中で対応するイベントが存在するものを表 1 に示す．

これらについては，発生したイベントに対応して，抽象動作を記録できる．対応するイベントのない動作については，静的解析から得られる情報も利用し，それぞれ以下のように取得する．

表 1 JDI のイベントと対応する抽象動作

抽象動作	JDI のイベント
Alloc	MethodEntryEvent (Constructor)
Call	MethodEntryEvent
Return	MethodExitEvent
Start	ThreadStartEvent
Load	ClassPrepareEvent
Lock	MethodEntryEvent (synchronized)
Unlock	MethodExitEvent (synchronized)
Throw	ExceptionEvent

Free Java では GC は実装依存となり，オブジェクトの明示的な消滅は存在しないため，プログラム終了時に一括で生成．

Refer フィールドの参照は AccessWatchpointEvent で取得．ローカル変数の参照イベントはないため，バイトコードレベルの StepEvent がローカル変数を参照するコードであった場合に生成．

Define フィールドへの代入は ModificationWatchpointEvent で取得．ローカル変数への代入イベントはないため，バイトコードレベルの StepEvent がローカル変数に代入するコードであった場合に生成．

Throws Throws は明示されないため，Throw から Catch(もしくは main method 終了) 節開始までの間に，抜けたメソッドの数だけ生成．

Catch イベントでは明示されないため，静的解析で，Catch 節の位置を特定し，その位置の最初のバイトコードの StepEvent で取得．

以上より，振る舞いモデルの全ての抽象動作を JDI のイベントで網羅することが可能である．抽象動作の列が構築できれば，記憶領域と値は，それぞれの抽象動作と対応するバイトコード命令のオペランドを JDI から取得可能である．その際，要素がモデル上に存在していない場合はその場で要素を作成し，存在している場合はその要素を特定し，抽象動作に対して関連を張る．

4. おわりに

本稿では，動的解析を用いる CASE ツールのためのプラットフォームのベースとなるプログラムの振舞いモデルについて紹介し，実際のプログラムの実行からモデルを構築するための情報取得部の構成について述べた．今後の課題は，本稿の情報取得部を用いて構築された振舞いモデルを利用するための API を準備し，プラットフォームが動的解析を用いる CASE ツールの作成に有効であることを示すことである．