6Q−5

プログラムに対する問い合わせシステムの構築 ～プログラムのXMLによる表現～

森崎　泰成†　　刈谷　丈治††　　田中　稔†††
†山口大学大学院理工学研究科　††山口大学総合情報処理センター　†††山口大学工学部

1．はじめに

計算機が普及するにつれ、ソフトウェアの需要は高まり、作成されるプログラムは膨大なものとなっている．プログラムの規模の拡大からプログラムの再利用が求められているが[1]、困難な場合が多い．プログラムからプログラマにとって必要な情報が得にくく、他人が作成したプログラムの内容をすぐに理解できないことが一因にあると考えられ、このことの解決が求められている．

本稿では、ソースプログラムを静的に解析してプログラマが要求する情報の提供を行い、必要であるならば、作成者の意図に関する問い合わせにも回答して、プログラムに対する理解を支援する、プログラムに対する問い合わせシステムについて述べる．

2．問い合わせシステム

問い合わせシステムは、プログラマのプログラムに関する情報の提示要求に対して回答するシステムである．プログラマは、解析対象とするソースプログラム群をシステムへ入力し、問い合わせたい項目を選択することにより、回答を得られる．

本システムの構成を以下に示す（図1）．

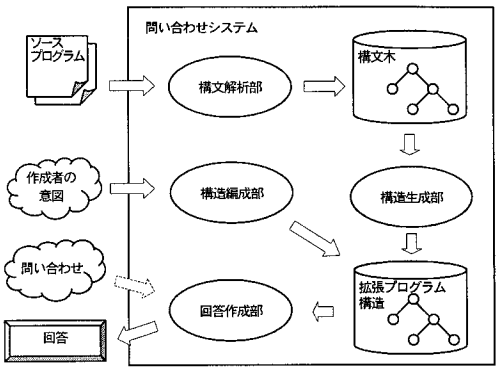

図1：問い合わせシステムの概要

問い合わせシステムは、以下の6部より構成される．

- 構文解析部：ソースプログラム群に対して、文法に沿って構文解析を行ない、構文木群を作成する．
- 構造生成部：得られた構文木から、検索に適した構造へ変換し、ノード間の関係など解析した結果を付加した拡張プログラム構造を作成し、データベースに蓄える．
- 回答作成部：拡張プログラム構造から、プログラマの選択した問い合わせに適した構造を作成し、その構造から回答を作成する．
- 構造編成部：プログラム作成者のプログラム言語上では表現できない意図について、特定の意図（後述）を拡張プログラム構造に埋め込む．
- 構文木データベース：個々のソースプログラムに対応した構文木をXMLにより蓄える．
- 拡張プログラム構造データベース：拡張プログラム構造をXMLにより蓄える．

本システムでは、問い合わせの対象をJava言語によるプログラムとした．また、XMLの利用によりシステム固有の構造ではなくなり、プログラマが内部構造に直接触れることが可能となる．構文解析にはANTLRを用い、静的な解析により得られる情報を取り扱う．

3．拡張プログラム構造

3．1　基本構造

Java言語プログラムに関する情報は、大きく分けると表1の49種類に分類される[2]．構造生成部では、構文木群のタグを表1の対応するタグへ変換する．このタグをひとつの構造とする．それぞれの構造に対して、名称や引数、包含構造など文法により得られる属性から、参照関係や宣言場所といった、解析した結果より得られる属性まで付加していく．得られた構造群を拡張プログラム構造の基本構造とする．

この基本構造から、個々の問い合わせに対して適当な構造を作成し、回答を行なう．また、構造にはXMLを用いているので、DOM等のAPIにより、プログラマがこれらの構造を用いて新たに構造を生成して、必要な情報を独自に得ることも可能である．

Development of the Inquiry System for Programs
~ Description of the Java Programs in XML ~
Taisei Morisaki†, Joji Kariya††, and Minoru Tanaka†††
†Graduate School of Science and Engineering, Yamaguchi University
††Integrated Information Processing Center, Yamaguchi University
†††Faculty of Engineering, Yamaguchi University

表1：プログラム情報の保持対象

プロジェクト	パッケージ	ファイル	クラス
インタフェース	内部クラス	内部インタフェース	静的初期化子
インスタンス初期化子	フィールド	コンストラクタ	メソッド
ブロック	局所変数宣言文	空文	ラベル付き文
if 文	switch 文	case 節	default 節
while 文	do 文	for 文	break 文
continue 文	return 文	throw 文	synchronized 文
try 文	catch 節	finally 節	インスタンス生成式
配列生成式	インスタンス呼出し式	配列アクセス式	単項演算式
二項演算式	三項演算式	代入文	仮引数
配列初期化子	リテラル	クラスリテラル	優先式
this	変数	コメント	
代替コンストラクタ this	代替コンストラクタ super		

3．2　基本構造の拡張（意図の埋め込み）

設計からプログラムを作成する過程で、プログラム言語で表現できないことは失われる．この損失を少なくすることにより、プログラムに対する理解が高められる．本システムでは、この損失を減らすために形式的に取り扱える特定の意図(表2)について、拡張プログラム構造に埋め込むことができる．

プログラム作成者がソースプログラム群をシステムへ入力した後、埋め込みたい意図に関する項目を選択し、必要な情報を入力すると、システムは入力された情報が正しいのかを検証する．検証した結果、正しければ意図情報ファイルを作成し出力する．問い合わせ時には、この意図情報ファイルをソースプログラム群と合わせてシステムへ入力することにより、作成者の意図を知ることができる．

表2：埋め込むことのできる意図

パッケージ	あるパッケージを使用できるパッケージを制限する
クラス	あるクラスを使用できるクラスを制限する
	あるクラスを拡張できるクラスを制限する
フィールド変数	あるフィールド変数に代入できる型を指定する
	あるフィールド変数を更新できるメソッドを制限する

4．問い合わせへの回答

回答作成部は、クラスの継承関係や、メソッドの呼出し関係等の問い合わせに回答する．そのうち、パッケージに関するアクセス制限の埋め込みと、その問い合わせの例を以下に示す．

複数のパッケージを作成している場合、その中には、あるパッケージのために作成したパッケージも含まれることがある．通常、そのようなパッケージは、他のパッケージから使用されることを想定していない．本システムを用いると、プログラム作成者は、「パッケージAを使用できるパッケージは、パッケージBだけである」と制限できる．プログラム利用者が、このパッケージ群を用いてプログラムを作成した場合、問い合わせ時に利用者とシステムは以下の動作を行なう．

1．プログラム利用者は、利用者自身が作成したプログラムとパッケージ作成者のパッケージ群と意図情報ファイルをシステムへ入力する．
2．システムは、入力から拡張プログラム構造を作成する．
3．プログラム利用者は、問い合わせ項目のひとつである「意図に沿ってプログラムができているか」という項目を選択する．
4．システムは、表1の構造から表2の意図が埋め込まれている構造を検索する．検索の結果、ある構造「パッケージ」に属性「使用可能パッケージ」が見つかるので、すべての構造「ファイル」の属性「使用パッケージ」と比較して検証を行なう．
5．検索の結果、発見した構造とその検証結果を提示する．

プログラム利用者は、この問い合わせによりパッケージ作成者の意図を確認でき、また、その意図に違反していないかを知ることができる．

5．おわりに

本稿では、プログラムへの理解を支援するために構築した、プログラムに対する問い合わせシステムについて述べた．本システムの利用により、第三者が作成したプログラムへの早期理解が得られると考える．

本システムでは、ソースプログラムを静的に解析するため、動的に生成されるクラスインスタンス等には対応できない．また、実行時の動作に関する問い合わせも多いことが考えられる．このため、動的な解析との組合せが必要になると考える．

プログラム作成者の意図を形式的に扱えるようにすることは、第三者にその意図を伝えるために有用であるが、作成者自身がその意図を整理するという意味においても、有用なことである．今後の予定として、現行システムで扱える作成者の意図の種類を拡張していく考えである．

6．参考文献

[1] 戸松 豊和: Java プログラムデザイン, ソフトバンク パブリッシング, 1998.
[2] 細木 孝治: プログラムに対する問い合わせに適したデータ表現の提案, 山口大学大学院理工学研究科 2000 年度修士論文, 2001.