5106

プログラム作成支援ツールのための一つの試み

An Approach to Programming Support Tools

EE05 岩崎 優
指導教員 吉村 晋

1. はじめに

ソフトウェアの大規模化,プログラミング言語の高級化に伴い,プログラムデバック,プログラムの理解は困難なものになりつつある.そのため,様々なプログラミング言語に対するプログラム解析手法の提案,及びツールの実装がなされている[1].

2. 研究目標

C#で書かれたソースコードを把握するサポートツールを試作し,参照関係やデータ構造を図として提供し,開発者を支援することを目的とする.

3. 研究アプローチ

ソースコード情報を取り出すには, コンパイラの字句解析,構文解析と意味解析を利用することが良い[2].まず字句解析により,ソースコードを最小単位のトークン(字句)に分割する. 字句解析結果を構文解析器に渡し,正しい構文のチェックを行う.

以上の結果を基に,XML 文書を作成する. XML 文書は,要素などのマーク付けで構造化され,ツリー構造を形成する.

XML は目的に応じて要素名や属性などを自由に定義できる. そこでソースコードの解析情報を持った XML 文書を GUI へ入力し, クラスダイアグラムなどの図を利用してユーザーへ提供する目標を立てた.

4. XML ジェネレーターの試作

解析対象となるソースコード例を Fig.1 に示す.また Fig.1 を解析して XML 文書として出力した結果をFig.2 に示す.出力された XML 文書は,基となる Fig.1 に示したソースコードの参照関係,入れ子構造,識別子などの情報を持つ.

```
using System;
namespace MySpace
{
        class SubClass
        {
                int x;
                public SubClass(int arg)
                {
                        x=arg;
                }
                public void Show()
                {
                        Console.WriteLine("x= {0}",x);
                }

        }
        class MyClass
        {
                public static void Main()
                {
                        int foo=500;
                        SubClass sc=new SubClass(foo);
                        sc.Show();
                }
        }
}
```

Fig.1 解析対象例

```
<?xml version="1.0" encoding="UTF-8"?>
<Text file="I:\卒研\sampleXML.xml">
  <using name="System" ID="10000"/>
  <Namespace name="MySpace" ID="10001">
    <class modifier="" name="SubClass" ID="10002">
      <value modifier="" name="x" type="int"  ID="10003"/>
      <constructor modifier="public" name="SubClass" ID="10004">
        <argument name="arg" type="int" ID="10005"/>
        <Operation ID="10006">
      <value name="x" ref="10003"/>
      <Equal/>
      <value name="arg" ref="10005"/>
        </Operation>
      </constructor>
      <method modifier="public" returntype="void" name="Show" ID="10009">
        <Execution name="Console.WriteLine" ID="10010">
          <argument>
            <StringLiteral>x=<value name="x" ref="10003"/></StringLiteral>
          </argument>
        </Execution>
      </method>
    </class>
    <class modifier="" name="MyClass" ID="1011">
      <MainMethod modifier="public" static="true" returntype="void" ID="10012">
        <Operation ID="10013">
          <value modifier="" name="foo" type="int" ID="10014"/>
          <Equal/>
          <IntegerLiteral>500</IntegerLiteral>
        </Operation>
        <Operation ID="10015">
          <value modifier="" name="sc" typeref="10002" ID="10016"/>
          <Equal/>
          <New/>
          <Execution name="SubClass" ID="10017" ref="10004">
            <argument name="foo" ref="10014"/>
          </Execution>
        </Operation>
        <Execution name="sc.Show" ID="10018" ref="10009"/>
      </MainMethod>
    </class>
  </Namespace>
</Text>
```

Fig.2 解析結果の XML 文書

5. 考察

上記の部分の試作は, 当初, 字句解析と構文解析を自動生成する JavaCC[3]を利用したが, 冗長で読みにくい Java コードが自動生成され, デバックが困難であった. そこで JavaCC の字句解析, 構文解析の機能を C#言語に移植し,改良した.

また,単一ソースコードファイルを解析対象としているため,複数ファイルの対応が必要になる.

6. おわりに

本研究では,C#言語のソースコードを対象とした字句解析器,及び構文解析器,並びにソースコードに対応した XML 文書を生成する XML ジェネレーターを作成した.現在,XML 文書を効果的に表示する GUI 部分を作成中である.

7. 文 献

[1] 山中 祐介,大畑 文明,井上 克朗, “プログラム解析情報の XML データベース化”, コンピュータソフトウェア,Vol.19,No1(2002),pp39-43,2001 年 8 月 10 日

[2] 疋田 輝雄,石畑 清,”コンパイラの理論と実現” 共立出版株式会社,2003 年 10 月 20 日

[3] 五月女 健治,”JavaCC コンパイラコンパイラ for Java” テクノプレス,2005 年 2 月 25 日