1. Goal

The goal of the project is to compile an arbitrary Eiffel program to Java Byte Code. The generated program should run on a standard Java Virtual Machine. A code generator is added to the ISE Eiffel Compiler to do the Job. A Java programer should be able to use this generated classes in his code in a straightforward way. In particular he should be able to inherit from such generated classes and use them as a client. The generated code should also not loose the information about the subtyping structure of the Eiffel program.

2 Overview

Multiple inheritance can be modeled in Java with the help of interfaces. The interfaces only reflect the subtyping relation of Eiffel classes. For every Interface a class is generated which implements that interface and contains the translated code from the corresponding Eiffel class. Since multiple subclassing is not possible in Java subclassing is replaced by delegation. An implementing class holds references to other implementing classes. This classes are superclasses in Eiffel.

3 Example

The very simple Eiffel program:

```eiffel
class CLASS_A
feature
  f is do σ0 end
end

class CLASS_B
feature
  g is do σ1 end
end

class CLASS_C
inherit
  CLASS_A
  CLASS_B
end

local
  class_c : CLASS_C
do
  create class_c
```
can be translated to Java like this:

```java
public interface ClassA {

    public void f();
}

public interface ClassB {

    public void g();
}

public interface ClassC extends ClassB, ClassA {

}

public class ClassAImpl implements ClassA {

    public void f() {
        \( \sigma_0 \)
    }
}

public class ClassBImpl implements ClassB {

    public void g() {
        \( \sigma_1 \)
    }
}

public class ClassCImpl implements ClassC {

    private ClassAImpl classADelegate = new ClassAImpl();
    private ClassBImpl classBDelegate = new ClassBImpl();

    public void g() {
        classBDelegate.g();
    }

    public void f() {
        classADelegate.f();
    }
}
```

Fig. 1: Multiple inheritance in Eiffel.

Benno Baumgartner – 4th April 2005
3. Example

```java
ClassC classC = new ClassCImpl();
classC.f();
classC.g();
```

Now the object referenced by classC is of type: ClassCImpl, ClassC, ClassA, ClassB and Object and a call to f() respectively g() will execute the correct code sequences.

Fig. 2: Multiple inheritance in Java.