Attribute Protection (Hidden Protected)

Jeg fant dette eksempelet i TestBatchen til Simula as.

```
01:   Class A;
02:   Begin
03:      integer i;
04:   end;
05:
06:   A Class B; Protected i;
07:   Begin
08:      integer i;
09:      procedure p; xb.i := 17;
10:   end;
11:
12:   ref(B) xb;
13:   xb :- new B;
14:
15:   xb.p;
16
17:   xb.i := 5;
18:
19:   inspect xb do i := 7;
```

Siden 'i' er Protected i klassen B vil den være usynlig ved remote access og ved inspection. Jeg tolker ihvertfall slik.

Alle tre eksemplene (linje 15-19) vil da aksessere 'i'en til A siden 'i'en til B er usynlig gjennom remote access.

Imidlertid; et av testprogrammene til Simula as. tolker linje 09 annerledes. Der forventes det at det er 'i'en til B som aksesseres.

Er det noe jeg har misforstått ?

Algoritme: Find remote attribute

```
//*****************************************************************************
// *** Utility: findRemoteAttributeMeaning
//*****************************************************************************
public Meaning findRemoteAttributeMeaning(String ident)
{return(findRemoteAttributeMeaning(ident,false)); }

public Meaning findRemoteAttributeMeaning(String ident,boolean behindProtected)
{ boolean prtected=false;
  for(String prct:protectedList)
     if(ident.equalsIgnoreCase(prct)) { behindProtected=prtected=true; break; }
  if(!prtected)
  { for(Parameter parameter:parameterList)
      if(ident.equalsIgnoreCase(parameter.identifier))
          return(new Meaning(VariableKind.parameter,parameter,this,this,behindProtected));
    for(Declaration declaration:declarationList)
        if(ident.equalsIgnoreCase(declaration.identifier))
          return(new Meaning(VariableKind.attribute,declaration,this,this,behindProtected));
    for(LabelDeclaration label:labelList)
        if(ident.equalsIgnoreCase(label.identifier))
          return(new Meaning(VariableKind.label,label,this,this,behindProtected));
    for(Virtual virtual:virtualList)
        if(ident.equalsIgnoreCase(virtual.identifier))
          return(new Meaning(VariableKind.virtual,virtual,this,this,behindProtected));
  }
  BlockDeclaration prfx=getPrefix();
  if(prfx!=null)
  { Meaning meaning=prfx.findRemoteAttributeMeaning(ident,behindProtected);
    if(meaning!=null) meaning.declaredIn=this;
    return(meaning);
  }
  return(null);
}
```

Hvor klassen Meaning ser slik ut:

```
public class Meaning {
       public VariableKind variableKind;
       public boolean foundBehindProtected;
       public Declaration declaredAs;
       public DeclarationScope declaredIn; // Search started here
       public DeclarationScope foundIn;    // Search ended here

        ... ...
}
```

I tilfellet Connection Block blir det litt annerledes. Vi må rette opp 'meaning' for å reflektere en Connection Block.

```
public Meaning findMeaning(String identifier)
{    Meaning result=classDeclaration.findRemoteAttributeMeaning(identifier);
     if(result!=null)
         result=new Meaning(VariableKind.connectedAttribute
                         ,result.declaredAs,this,result.foundIn
                         ,result.foundBehindProtected);
     else if(declaredIn!=null) result=declaredIn.findMeaning(identifier);
     if(result==null) Util.error("Undefined variable: "+identifier);
     return(result);
}
```