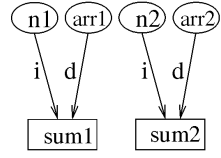


```

procedure Sum1_and_Sum2
  ( n1, n2 : integer;
    arr1, arr2 : int_array;
    var sum1,
        sum2 : integer );
var i : integer;
begin
  sum1 := 0;
  sum2 := 0;
  for i := 1 to n1 do
    sum1 := sum1 + arr1[i];
  for i := 1 to n2 do
    sum2 := sum2 + arr2[i];
end;

```



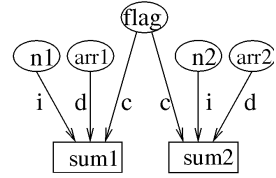
Coincidental cohesion

(a)

```

procedure Sum1_or_Sum2
  ( n1, n2, flag : integer;
    arr1, arr2 : int_array;
    var sum1,
        sum2 : integer );
var i : integer;
begin
  sum1 := 0;
  sum2 := 0;
  if flag = 1
    for i := 1 to n1 do
      sum1 := sum1 + arr1[i];
    else
      for i := 1 to n2 do
        sum2 := sum2 + arr2[i];
end;

```



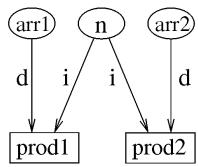
Conditional cohesion

(b)

```

procedure Prod1_and_Prod2
  ( n : integer;
    arr1, arr2 : int_array;
    var prod1,
        prod2 : integer );
var i : integer;
begin
  prod1 := 1;
  prod2 := 1;
  for i := 1 to n do begin
    prod1 := prod1 * arr1[i];
    prod2 := prod2 * arr2[i];
  end;
end;

```



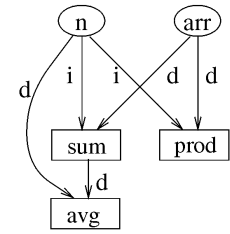
Iterative cohesion

(c)

```

procedure Sum_and_Prod
  ( n : integer;
    arr : int_array;
    var sum,
        prod : integer;
    var avg : float );
var i : integer;
begin
  sum := 0;
  prod := 1;
  for i := 1 to n do begin
    sum := sum + arr[i];
    prod := prod * arr[i];
  end;
  avg := sum / n;
end;

```



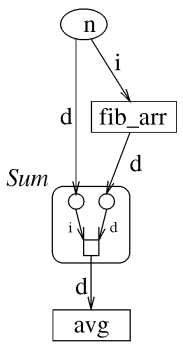
Communicational cohesion

(d)

```

procedure Fibo_Avg
  ( n : integer;
    var fib_arr : int_array;
    var avg : float );
var sum : integer;
    i : integer;
begin
  fib_arr[1] := 1;
  fib_arr[2] := 2;
  for i := 3 to n
    fib_arr[i] := fib_arr[i-1]
      + fib_arr[i-2];
  Sum(n, fib_arr, sum);
  avg := sum / n;
end;

```



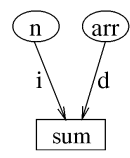
Sequential cohesion

(e)

```

procedure Sum
  ( n : integer;
    arr : int_array;
    var sum : integer );
var i : integer;
begin
  sum := 0;
  for i := 1 to n do
    sum := sum + arr[i];
end;

```



Functional cohesion

(f)

Fig. 1. IODG and DLC levels for six simple procedures.