CLU Examples

• From Liskov et al, CACM Aug77 article
  Abstraction Mechanisms in CLU
    - EG1 - how to define a class in CLU; shows rep type and operation signatures but not the specification
    - EG2 - how to define an iterator in CLU

Fig. 3. The wordbag cluster.

```
wordbag = cluster is
  create, % create an empty bag
  insert, % insert an element
  print; % print contents of bag
  rep = record [contents: wordtree, total: int];
  class interface

create = proc ( ) returns (cv);  % Rep type
         return (rep$[contents: wordtree$create ( ), total: 0]);
         end create;

insert = proc (x: cv, v: string);
         x.contents := wordtree$insert (x.contents, v);
         x.total := x.total + 1;
         end insert;

print = proc (x: cvl, o: outstream);
         wordtree$print (x.contents, x.total, o);
         end print;

end wordbag;
```

Data abstraction is wordbag:
Rep type is record - binary tree
Outside cluster only reveal wordbag
Within cluster see implementation of operations in terms of rep type - binary tree record
oneof: A union type where value depends on whether or not wordtree is empty.

Tagcase stmt: depends on rep type of current wordtree

Iterator: to separate selection of next object from a collection and performing an operation on that object. Provides objects from a collection one-by-one without repeats.

Fig. 5. Use and definition of a simple iterator.

count_numeric = proc (s: string) returns (int);
  count: int := 0;
  for c: char in string chars (s) do
    if char.is_numeric (c)
      then count := count + 1;
    end;
  end;
  return (count);
end count_numeric;

string_chars = iter (s: string) yields (char);
  index: int := 1;
  limit: int := string$size (s);
  while index <= limit do
    yield (string$f (s, index));
    index := index + 1;
  end;
end string_chars;